P-Series Installation and Operation Guide

Version 2.3.1.2   May 27, 2008   PN: 100-00055-01

FORCE10™

Copyright 2008 Force10 Networks®

Trademarks

Force10 Networks® and E-Series® are registered trademarks of Force10 Networks, Inc. Force10, the Force10 logo, and P-Series are trademarks of Force10 Networks, Inc. All other brand and product names are registered trademarks or trademarks of their respective holders.

Statement of Conditions

In the interest of improving internal design, operational function, and/or reliability, Force10 Networks reserves the right to make changes to products described in this document without notice. Force10 Networks does not assume any liability that may occur due to the use or application of the product(s) described herein.

USA Federal Communications Commission (FCC) Statement

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designated to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy. If it is not installed and used in accordance to the instructions, it may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to take whatever measures necessary to correct the interference at their own expense.

Properly shielded and grounded cables and connectors must be used in order to meet FCC emission limits. Force10 Networks is not responsible for any radio or television interference caused by using other than recommended cables and connectors or by unauthorized changes or modifications in the equipment. Unauthorized changes or modification could void the user’s authority to operate the equipment.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Canadian Department of Communication Statement

The digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Attention: Le present appareil numerique n’ emet pas de perturbations radioelectriques depassant les normes applicables aux appareils numeriques de la Class A prescrites dans le Reglement sur les interferences radioelectriques etabli par le ministere des Communications du Canada.

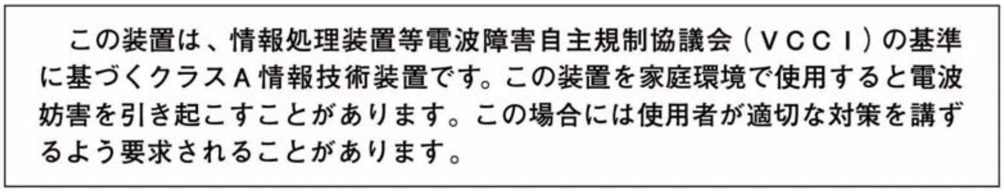
European Union EMC Directive Conformance Statement

This product is in conformity with the protection requirements of EU Council Directive 89/336/EEC on the approximation of the laws of the Member States relating to electromagnetic compatibility. Force 10 Networks can not accept responsibility for any failure to satisfy the protection requirements resulting from a non-recommended modification of this product, including the fitting of non-Force10 option cards.

This product has been tested and found to comply with the limits for Class A Information Technology Equipment according to CISPR 22/ European Standard EN 55022. The limits for Class A equipment were derived for commercial and industrial environments to provide reasonable protection against interference with licensed communication equipment.

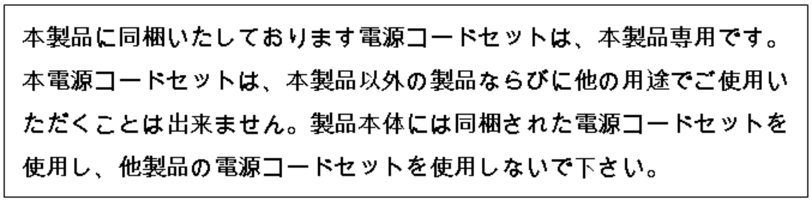
⚠ **Warning:** This device is a Class A product. In a domestic environment, this device can cause radio interference, in which case, the user may be required to take appropriate measures.

VCCI Compliance for Class A Equipment (Japan)



この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

This is Class A product based on the standard of the Voluntary Control Council For Interference by Information Technology Equipment (VCCI). If this equipment is used in a domestic environment, radio disturbance may arise. When such trouble occurs, the user may be required to take corrective actions.



本製品に同梱いたしております電源コードセットは、本製品専用です。本電源コードセットは、本製品以外の製品ならびに他の用途でご使用いただくことは出来ません。製品本体には同梱された電源コードセットを使用し、他製品の電源コードセットを使用しないで下さい。

⚠ **Danger:** AC Power cords are for use with Force10 Networks equipment only, do not use Force10 Networks AC Power cords with any unauthorized hardware.

Feedback on Documentation?
Send email to techpubs@force10networks.com

Contents

Appendix E

Appendix F

Preface About this Guide

Objectives

This document provides installation and operation instructions for the P-Series P10 appliance.

Audience

This guide is intended to be used by network engineers. The P10 is a Unix-based product that runs rule management software based on Linux and FreeBSD. As such, understanding how to operate the appliance requires a basic knowledge of Unix, including the *vi* editor.

Conventions

This document uses the following conventions to describe command syntax:

Convention	Description
keyword	Keywords are in bold and should be entered at the command prompt as listed.
parameter	Parameters are in italics and require a number or word to be entered at the command prompt.
{X}	Keywords and parameters within braces must be entered at the command prompt.
[X]	Keywords and parameters within brackets are optional.
x|y	Keywords and parameters separated by a bar require you to choose one.

Information Symbols

Symbol	Warning	Description
⚡	Danger	This symbol warns you that improper handling and installation could result in bodily injury. Before you work on this equipment, be aware of electrical hazards, and take appropriate safety precautions.
⛔	Caution	This symbol informs you that improper handling and installation could result in equipment damage or loss of data.
⚠	Warning	This symbol informs you that improper handling could reduce your component or system performance.
➡	Note	This symbol informs you of important operational information.

Related Documents

Additional P-Series documentation is available on the software CD that came with the appliance and in the documentation section of the Force10 website, www.force10networks.com.

- *P-Series Release Notes*

Additional Resources

- Cox, Kerry and Gerg, Christopher. 2004. Managing Security with Snort and IDS Tools. Sebastopol, California: O’reilly Media, Inc.
- Snort.org. http://www.snort.org/

Chapter 1 Installation

Figure 1 P-Series P10 Appliance (Front View)

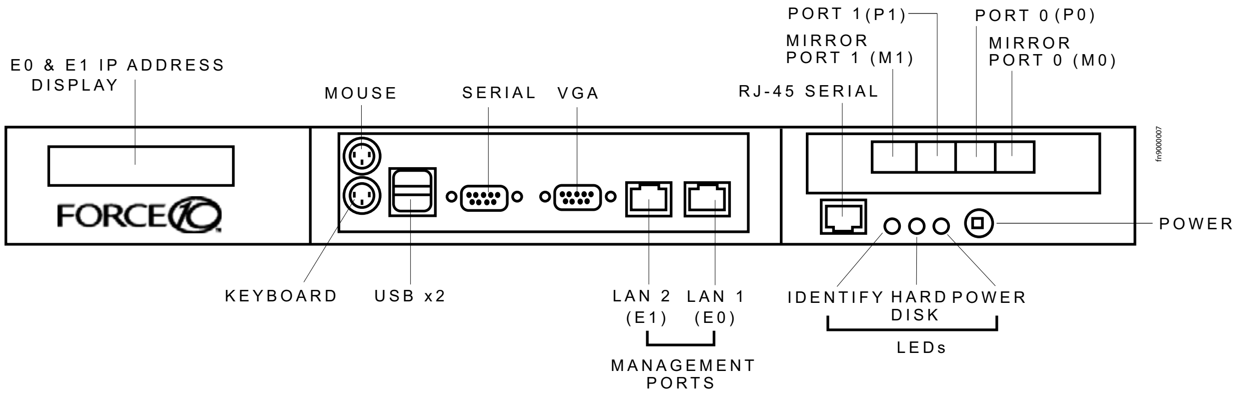
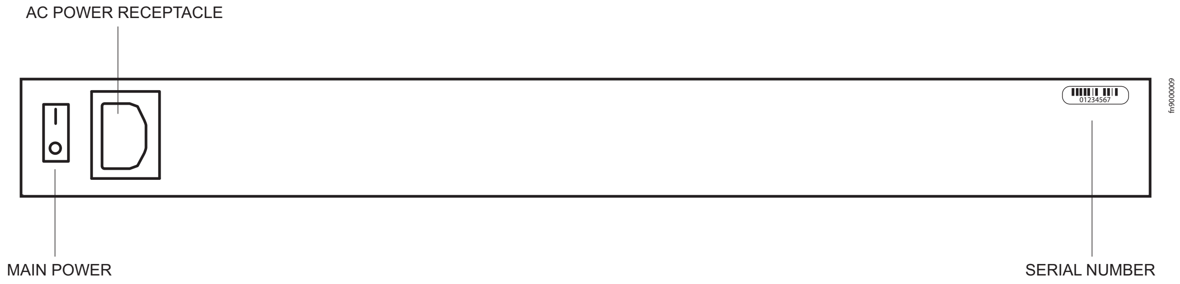


Figure 2 P-Series P10 Appliance (Rear View)



Label	Description
(LCD screen)	The LCD screen displays the IP address of the appliance next to either “e0:” or “e1:”, which represent LAN ports 1 and 2, respectively.
Port 1, Port 0	These two ports are sensing ports through which traffic is forwarded. They accept 10G XFP modules.
(unlabeled RJ-45 serial port next to IDENTIFY)	This port is not used.
IDENTIFY	This LED is not used.
HDD	This LED is blue when the hard disk is accessed.
PWR	This LED is green when the power is on.

Label	Description
(Power Button)	This button turns the appliance on and off. Press and hold the button to turn off the appliance.
(Laser Warning) CLASS 1 LASER PRODUCT LASERPRODUKT DER KLASSE 1 FN00048A	This label in the bottom right corner of the appliance indicates that the appliance is a Class 1 laser product that emits invisible laser radiation. This product complies with CDRH, 21 CFR 1040.

System Specifications

The specifications in Table 1 apply to the P-Series P10 appliance, Force10 catalog number PB-10GE-2P.

Table 1 System Specifications

Power	**AC Power Supply**	**Power Consumption**: 400W maximum, 260W nominal **Current**: 3.6 A @ 120V, 2.0 A @ 240V **Voltage**: 100-240V, 47-63Hz, 8A maximum input current **Heat Dissipation**: 1360 BTU/hr maximum, 888 BTU/hr nominal
	Battery	3V CR2032 coin cell
Physical	**Dimensions**	**Height**: 1.75 in **Width**: 17.6 in **Depth**: 15.5 in (1RU half depth)
	Weight	20 lbs (9.07 kg)
Environmental	**Temperature**	**Operating**: 41° to 104°F (5° to 40°C) **Storage**: -40° to 149°F (-40° to 65°C) **Relative humidity**: 20-80% (non-condensing)
	Altitude	**Operating**:-50 to 10,000 ft (-16 to 3048 m) **Storage**: -50 to 35,000 ft (-16 to 10,600 m)

Physical Connections

Note: Connections to the sensing, mirroring, and management ports require straight-through CAT5 cables.

Warning: Do not hot-swap XFPs. If they are accidentally removed, turn off the appliance, replace the XFPs, and then turn the appliance back on.

Step	Task
1	Review the system specifications and ensure that your operating and storage conditions meet the stated requirements.
2	Connect the power cable, a keyboard, and a monitor to the appliance.
3	Connect the LAN 1 port on the appliance to the local area network where DHCP is available. If a DHCP server is not available, an IP address can be assigned manually; see "Configuration" on page 12.
4	Install XFPs in the ports that will be used.
5	Connect the sensing ports to the devices from which the appliance will receive traffic. • Traffic originating from the device connected to Port 0 has Channel 0's rules applied to it. • Traffic originating from the device connected to Port 1 has Channel 1's rules applied to it.
6	(Optional) Connect the mirroring ports to the devices that will receive mirrored traffic. • Mirror Port 0 mirrors matched traffic from Channel 0. • Mirror Port 1 mirrors matched traffic from Channel 1.
7	Connect the power cable to a power source, and switch on the main power on the back of the appliance.
8	Press the power button on the front of the appliance to turn on the device.

Booting

During booting you can select the OS of your choice.

The management ports are configured for DHCP and probe for an IP address, gateway, and name server. The IP address is displayed on the LCD screen.

When the appliance is powered up, all packets are forwarded between its ports by default until the firmware and device drivers are loaded. Once they have been loaded, the DPI generates interrupts to the host processor and offers the captured packets in the same way as a standard network interface card in promiscuous mode.

Configuration

Once the appliance is booted:

Step	Task
1	Log in as root with the password **plogin**.
2	Change the password, if desired, with the command **passwd**.
3	Set the clock for the appropriate timezone using the command **tzsetup**. This command calls a graphical user interface that instructs you on how to select the appropriate timezone.

Security Check

The P10 is remotely accessible only via Secure Shell Daemon (SSHv1 or SSHv2). However, inspect the configuration, and make sure it meets the security policy requirements of your network before deploying the appliance.

Upgrading Software

Upgrading software requires a boot firmware (PROM) upgrade. This upgrade must be done during a maintenance window. During this period, stop all traffic from flowing through the appliance, and disconnect all cables from the XFPs.

➡ **Note:** You must be logged in as root to upgrade software.

Warning: Stop all traffic from flowing through the appliance, and disconnect all cables from the XFPs before proceeding.

Step	Task	Command
1	Save earlier configuration files and firmware by copying the directory */usr/local/pnic* to the home directory.	**cp -Rf /usr/local/pnic/ /home**
2	Create a new sub-directory in the home directory for the upgrade package.	**mkdir ~/***upgrade_directory*
3	From the root directory, secure copy the file *filename* from a server to the upgrade directory you created. **Note:** In Unix, the tilde symbolizes the home directory, and can be used in place of the absolute path to the home directory. The upgrade file is a Unix tarball, the file extension of which is *.tar.gz*.	**scp** *username@server:absolute_path/filename* **~/***upgrade_directory*
4	Change directory to upgrade directory you created.	**cd** *upgrade_directory*
5	Untar the file *PTPS-P_MAIN*.	**tar xvzf PTPS-P_MAIN**
6	Change directory to *SW*.	**cd SW**
7	Enter the command **gmake erase** followed by **gmake**.	**gmake erase** **gmake**
8	Enter the command **gmake install**.	**gmake install**
9	Verify that the new software version is installed.	**pnic cardstatus**
	Warning: The remainder of this procedure is for upgrading the boot firmware. The boot firmware upgrade process takes up to 30 minutes and must not be interrupted. If the process is interrupted, the boot firmware must be reloaded via JTAG, which requires an RMA.	
10	Enter the command **pnic loadeproms** to upgrade the boot firmware. Answer “yes” to the confirmation question. **Note:** This process takes up to 30 minutes.	**pnic loadeproms**
11	Reboot the appliance. **Note:** Reboot the appliance only after **pnic loadeproms** has successfully finished.	**shutdown -r now**
12	Log into the appliance and enter the command **pnic cardstatus**. Verify that there is an output for this command. This indicates that the upgrade process has been completed successfully. **Note:** See Appendix A , on page 79 for an example output for this command.	**pnic cardstatus**

Step	Task	Command
13	Re-compile all rules firmware with the new compiler located in the directory *pnic-compiler.*	**cd** *upgrade_directory*/**pnic-compiler** **gmake**
14	Install pre-compiled firmware if needed.	**cd** *upgrade_directory*/**firmware** **gmake install**

Chapter 2 Getting Started

To begin inspecting and filtering traffic you must:

1. Select firmware and dynamic rules
2. Set capture/forward policies
3. Check for proper operation by generating traffic across the appliance.

Step	Task
1	As root, enter the command **pnic gui** from the Unix command line to invoke a graphical user interface (GUI).
2	Enter the command **m** from the GUI command line.
3	Select **Manage Firmware** from the Rule Management GUI, then select "null" firmware and confirm. The sample firmware and rules files are testing examples only. Force 10 recommends not employing the sample firmware for production IDS/IPS use.
4	Select **Edit Rules** from the Rule Management GUI.
5	Uncomment the rule **alert on all icmp any any -> any any (msg:"@icmp";)** by removing the # symbol before the rule. • Enter the command **i** to enter insert mode. • Navigate to the character using the arrow keys, and delete the character.
6	Enter the command **:wq** to exit the *vi* editor, and confirm your changes.
7	Confirm to reload the Forward/Block settings.
8	Run a packet sniffer such as *tcpdump* on the network interface associated with the appliance.
9	Generate some ICMP traffic to be exchanged between endpoints. • *Endpoints* are two network nodes on opposite sides of the appliance such that traffic between those nodes passes through the appliance. • For example, enter **ping** ***destaddress***, where ***destaddress*** is the IP address of the endpoint on the opposite end of the appliance.
10	If you are using *tcpdump*, enter the command **tcpdump -i pnic0 -n** from the Unix command line. • This prints to standard output all of the packets captured by the DPI. • If the appliance is operating correctly, you will see the ICMP packets.

Returning to the Default Configuration

Return to the factory default settings using the command **pnic resetconf**. See the Command Line Reference, on page 79.

Chapter 3 Introduction

The P-Series P10 *Intrusion Detection and Prevention System* (*IDS/IPS*) appliance employs *Dynamic Parallel Inspection* (*DPI*) technology. It uses a Multiple Instruction Single Data (MISD) massively parallel processor that executes thousands of security policies or traffic capture operations on the same data stream at the same time.

DPI synthesizes individual security policies and packet analysis algorithms and maps them directly into silicon hardware "gates." Through this design it is able to deliver full packet inspection and protection at line rate for 1-Gigabit and 10-Gigabit links whether the traffic load or security policy is 1% or 100%.

The policies can be derived from public domain signatures, or they can be completely user-defined. For each policy, you can direct the DPI to:

- Capture packets for the host (capture is defined as both DMA to host and copying to the mirror port)
- Forward packets (with negligible delay)
- Block packets

As a result, the P10 can be used as both an IDS accelerator and a stateful content filter for IPS applications. In an active configuration, it can be inserted inline into the network; this alleviates the need for a SPAN port or tap and enables filtering applications. In passive configurations, it can merely listen to the network via a mirroring port or tap.

Hardware Architecture Overview

The P10 is a 1-RU appliance provisioned with one DPI processing system, and has at minimum: an AMD Dual Core Opteron 280 processor, a 400-GB hard drive, 8 GB of RAM.

Figure 3 shows packet flow in the DPI, which is a two-port device. Packets are forwarded from the receive side of the first port (Rx0) to the transmit side of the second port (Tx1). Likewise, Rx1 forwards packets to Tx0 of the first port.

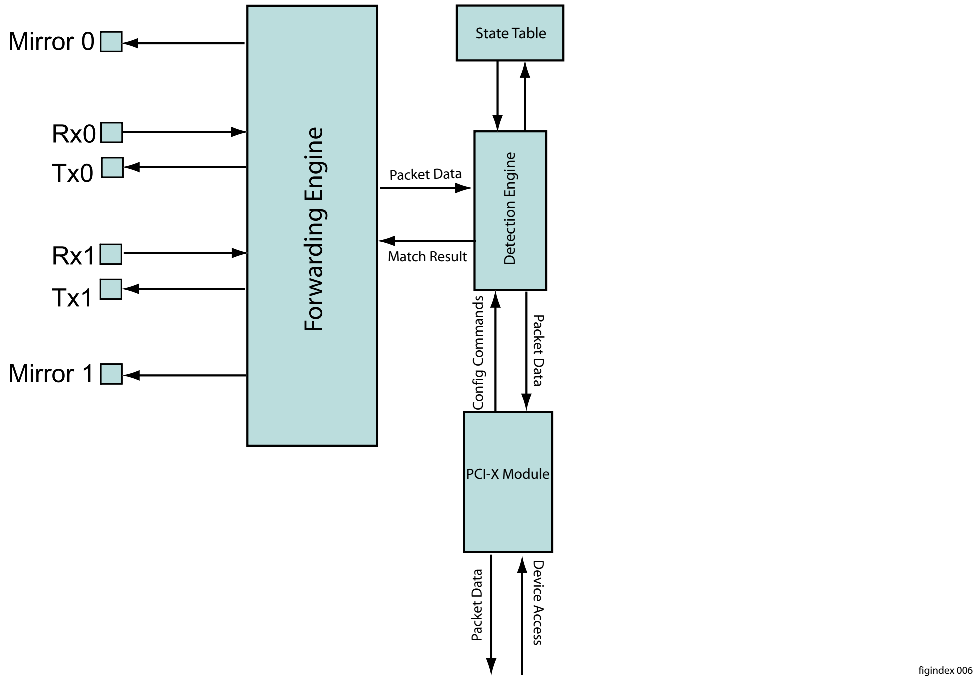
As the packets are being forwarded they are also processed in real time by two independent processing channels, each with its own set of policies. If there is a match in a processing channel, the DPI can block the packet, capture it, and send it to the host through the PCI-X bus. The two processing channels are completely independent, and thus they can be used to process two asymmetric links, or both directions of a full-duplex connection.

In addition to two sensing interfaces, the P10 includes two 1-Gigabit Ethernet mirroring ports. These ports can copy and forward matched traffic to another device. It is also possible to disable the PCI-X DMA capture, and let the matched traffic bypass the host entirely for applications in which host capture is not desired.

Figure 3 illustrates how all matched packets are copied and transmitted by mirror ports.

Note: Mirroring is automatically enabled when the mirroring port is connected to another network device. Mirroring is not controlled through the CLI.

Figure 3 Logic Diagram of Traffic Flow in the P10 DPI



Types of Rules

Two types of rules can be uploaded to the FPGA:

- *Static rules*: Static rules are compiled to become part of the firmware and are mapped directly into logic gates. Static rules can be set to capture/not capture and block/not block individually, but they cannot be changed once they have been loaded into the FPGA.
- *Dynamic rules:* Dynamic rules are programmed at runtime in the DPI hardware registers and can be configured without changing the firmware. These rules (like static rules) can be disabled/enabled individually.

Sample Rules and Firmware

The P10 includes sample rules files in the *pnic-compiler/rules* directory. You can browse these files in order to become more familiar with Snort syntax or creating rules files; you can also generate firmware from these files at your discretion.

Firmware is a set of rules that has been transformed—using a compiler—from Snort syntax into a form suitable for uploading to the FPGA. Two sets of sample rules files have been compiled into firmware and are available to be uploaded to the FPGA using either of two firmware management methods (see “Rule Management” on page 19). Table 2 describes each sample rules file.

Table 2 Sample Rules Files

Rule Set	Description
evasion.rules	The rules in this file help detect attacks which are using strategic TCP segmentation to avoid detection.
fw.rules	This file contains rules written in Snort syntax for a firewall application (see “Writing Rules for a Firewall Deployment” on page 77).
meta.rules	The rules in this file report on flow information and provide compatibility with Snort.
null.rules	This file contains no rules; the firmware created from these files are empty images that maximize the dynamic rule capacity (see “Rules Capacity” on page 55).
sample.rules	This file contains rules written in Snort syntax that were derived from publicly available IDS rules.

The firmware based on the sample rules files follow the naming convention described in “Selecting Firmware with the GUI” on page 30.

Note: Force 10 recommends not using the sample firmware for production IDS/IPS use. The sample firmware requires considerable site-specific customization in order to be effective; they are included only for you to become more familiar with the functionality of the appliance.

Rule Management

The P-Series software provides three methods by which you can manage the rules and functionality of the appliance:

- *Graphical User Interface*: The graphical user interface (GUI) is a menu-based method for managing the appliance.
- *Web-based GUI*: Manage the appliance and graphically plot performance online.
- *Command Line Interface*: The command line interface (CLI) uses a script called *pnic* through which you can manually perform the same management tasks as the GUI by entering commands at the command prompt.

Force10 recommends using the GUI or web-based GUI if no programmatic interface is required.

Deploying the P-Series

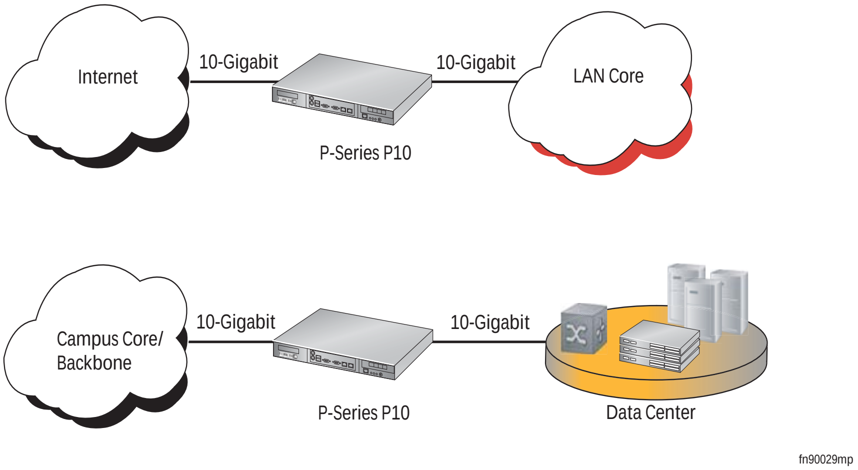
The flexible architecture of the P-Series lends itself to various deployments.

Inline Deployment

Use the P-Series for inline traffic inspection in IPS or firewall applications at 10-Gigabit line rate (Figure 4).

- For IPS deployment, no special configuration is needed; the P-Series is in inline IPS mode by default.
- For a firewall deployment, enable drop mode (see Command Line Reference on page 79).

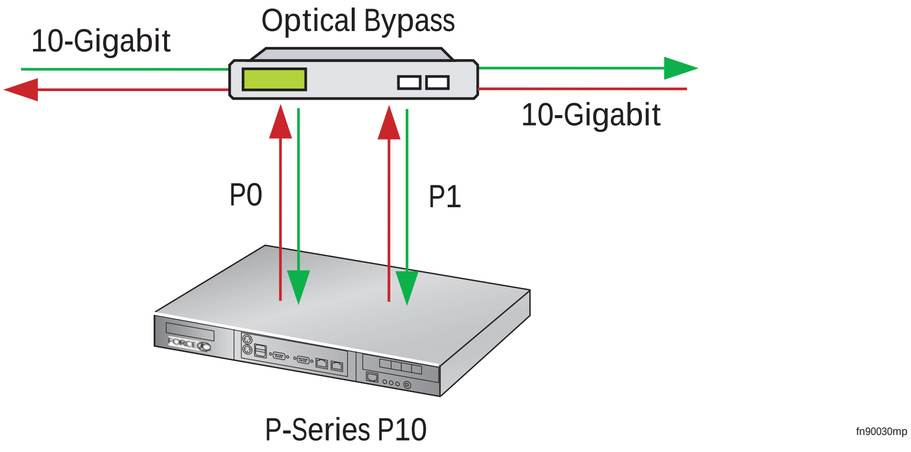
Figure 4 P-Series Inline Deployment



Fail-safe Deployment

The P-Series hardware is fail-safe. In the event of a software exception or reboot, the card continues to function as it did before the event. In the event of a power failure, the hardware stops functioning, and traffic is dropped. When the appliance powers up again, all the traffic is allowed by default, and the card functions as before. Use an optical bypass switch in an inline deployment so that traffic continues to flow in the event of a power failure, as shown in Figure 5.

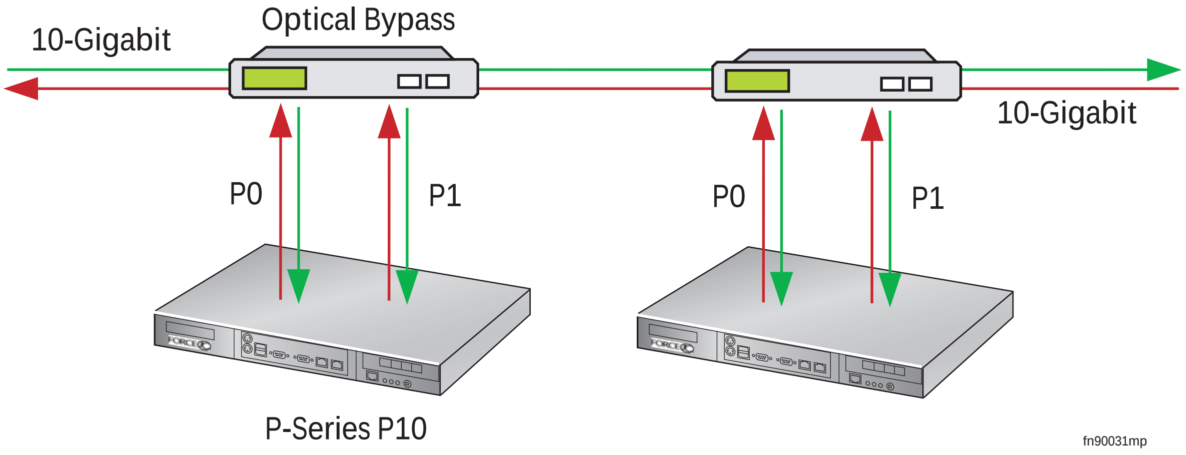
Figure 5 Fail-safe Behavior with Optical Bypass



Highly-available Deployment

Use optical bypass switches with the P-Series for a highly-available, redundant deployment, as shown in Figure 6. Both the appliances have the same configuration so that in the event of a power failure on one device, the other continues to operate, and the detection engine remains intact. In the event that both devices experience a power failure, the traffic continues to flow through the bypass switches.

Figure 6 Highly-available Redundant Deployment



Passive Deployment

Enable passive mode (see Command Line Reference on page 79) with fiber taps in line for IDS deployments.

- Send traffic from one side of the tap to port P0 and traffic from the other side to port P1, as shown in Figure 7.
- Aggregate traffic from both sides of the link to one port, as shown in Figure 8.
- Aggregate traffic from both sides of the link to one port using a SPAN port, as shown in Figure 9.

Figure 7 Passive Deployment using a Network Tap

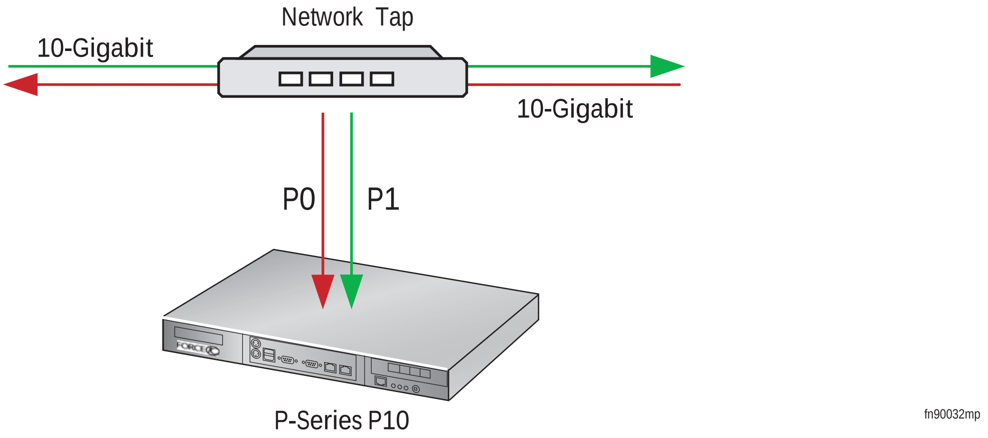


Figure 8 Passive Deployment with Aggregation using a Network Tap

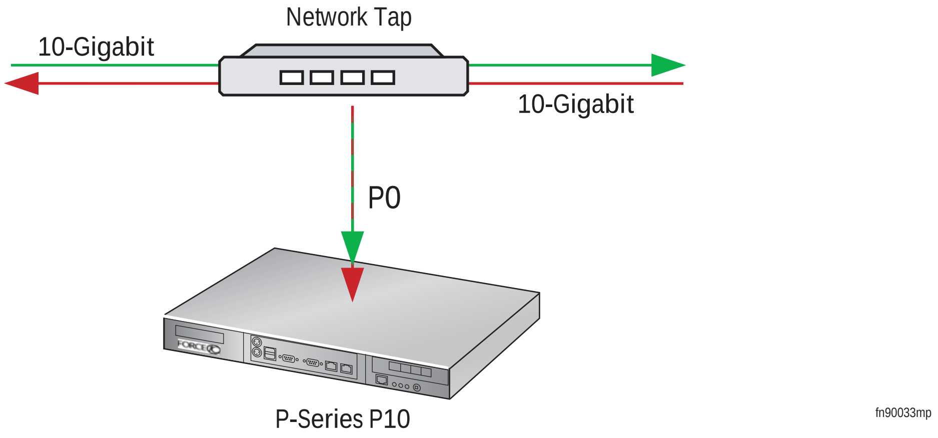
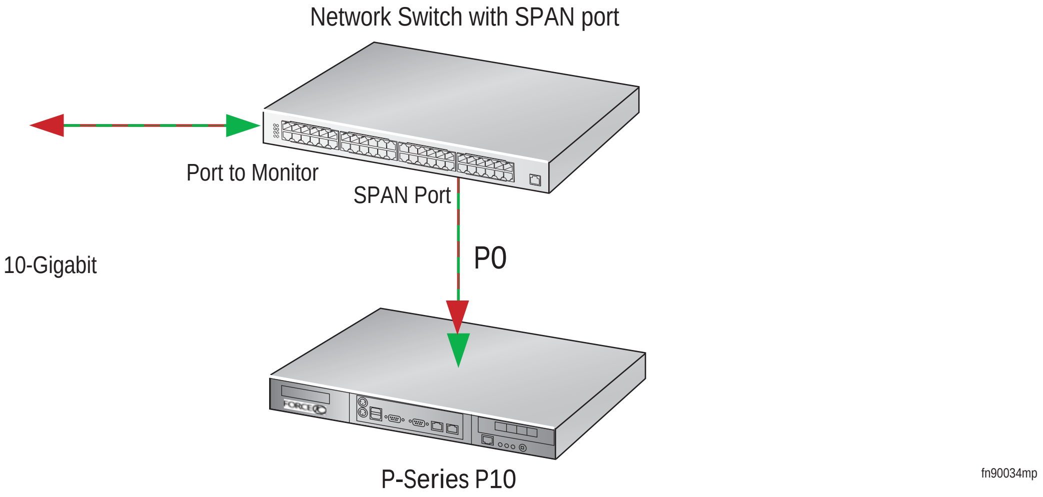


Figure 9 Passive Deployment with Aggregation using a SPAN port



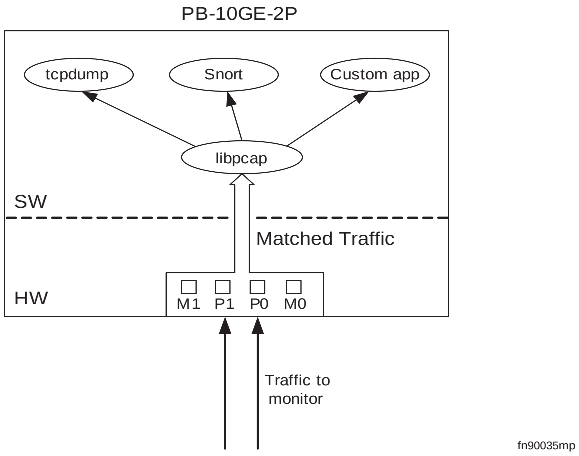
Capturing Matched Traffic

P-Series supports capturing matched traffic for analysis.

Capturing to a Host CPU

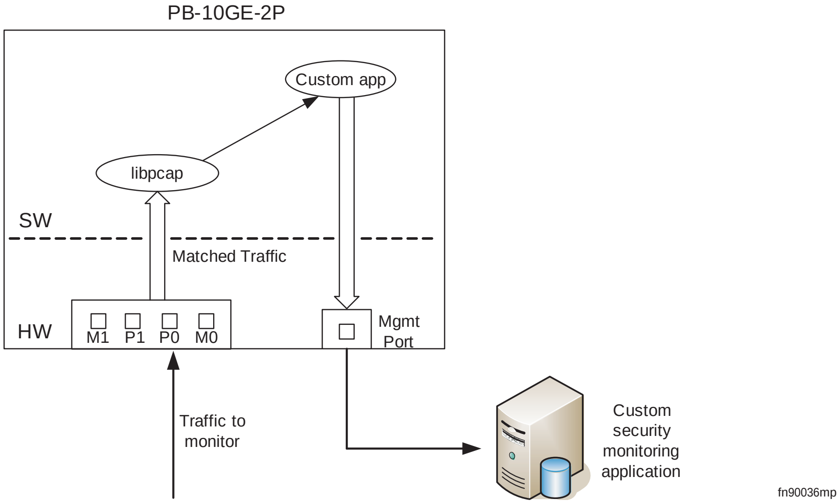
Captured traffic can be sent to a host CPU through a libpcap library interface, where it can be made available to applications for analysis. A typical implementation provides IDS/Snort acceleration because of the hardware assist.

Figure 10 Capturing Matched Traffic via the libpcap Interface



Use the P-Series in an integrated security monitoring solution through the management port. The P-Series comes with support for Sguil NSM (see Network Security Monitoring on page 43).

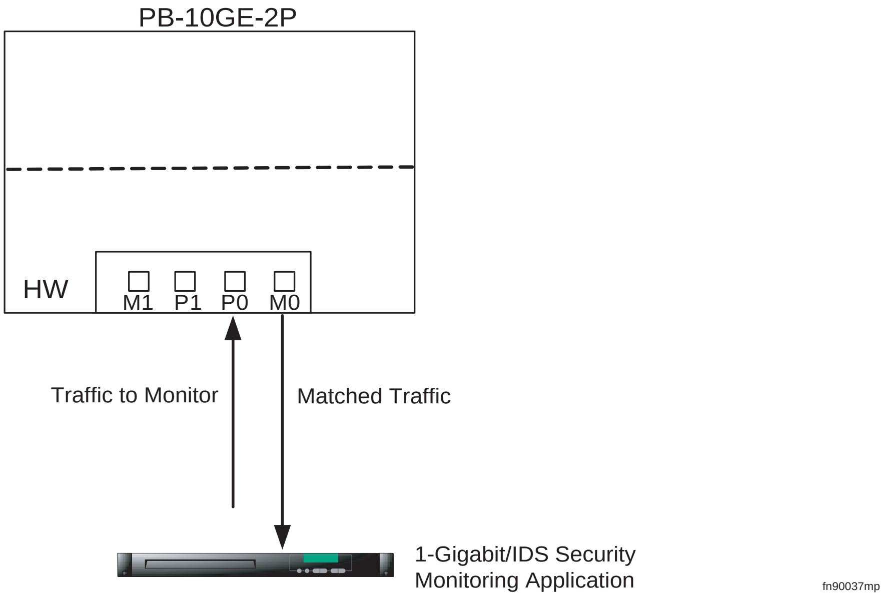
Figure 11 Creating a Network Monitoring Solution with the P-Series



Mirroring to Another Device

Mirror captured traffic out of the 1-Gigabit mirroring ports to use the P-Series as an IDS accelerator or as part of an integrated security monitoring solution.

Figure 12 Creating an IDS Accelerator with the P-Series



Chapter 4 Graphical User Interface

The GUI can be used to:

- Start and stop the DPI
- Load firmware
- Compile and load dynamic rules
- Manage the runtime parameters
- Manage the capture/forward policies for rules

➡ **Note:** Using the GUI requires the super user privilege.

To invoke the GUI:

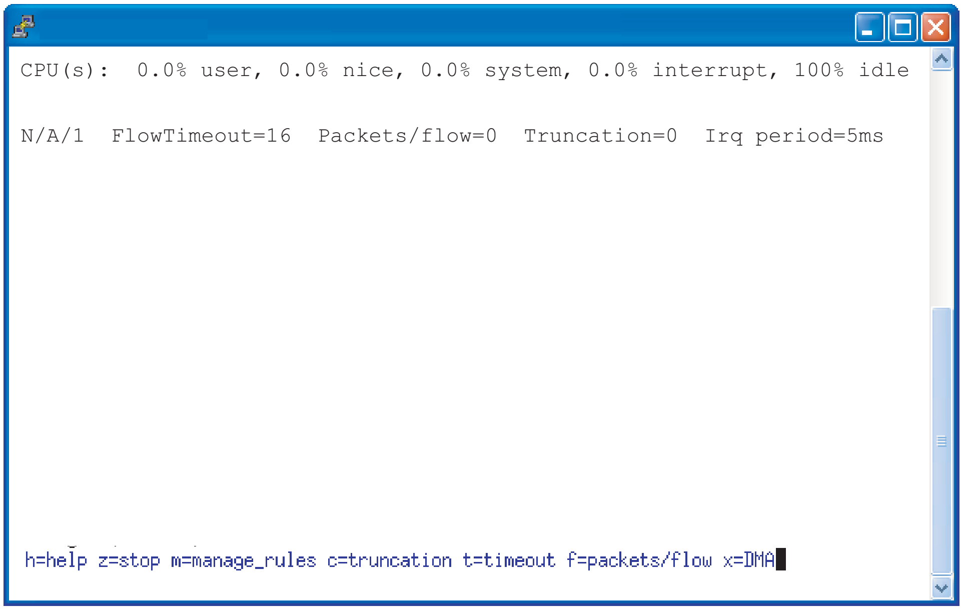
Step	Task
1	Invoke the GUI by entering the command **pnic gui**. **Note:** The OS environment variables are set such that the **pnic gui** command can be executed from any path.

Runtime statistics are displayed after the **pnic gui** command is executed. If the FPGA is not loaded, the display appears as shown in Figure 13. If firmware is loaded, the display appears as in Figure 19.

GUI Commands

From the Runtime Statistics display, you can enter commands to control the DPI (see Table 3, or enter the **h** command from the GUI command line).

Figure 13 Runtime Statistics - FPGA Unloaded



```
CPU(s):  0.0% user, 0.0% nice, 0.0% system, 0.0% interrupt, 100% idle

N/A/1  FlowTimeout=16  Packets/flow=0  Truncation=0  Irq period=5ms
```

```
h=help z=stop m=manage_rules c=truncation t=timeout f=packets/flow x=DMA
```

Note: GUI commands that require a subsequent value entry have the current value displayed in parentheses at the prompt.

Table 3 GUI Commands

Command	Description
a	Establishes the IRQ period (measured in milliseconds), which moderates DPI access to the PCI-X bus. Valid values are 1 to 255, where 1 is no throttling, and 255 is maximum throttling.
c	This command is not supported.
d	Brings the OS network interface down and disables matching.
f	Establishes the maximum number of packets to be captured for each flow (Packets/Flow). A value of 0 specifies all packets.
h	Displays help information about the commands.
i	Establishes the display refresh interval (measured in seconds).
m	Invokes a dialog menu through which dynamic rules can be defined, capture/forwarding policies can be set for each individual rule, and the firmware can be selected and loaded.(see Figure 14).
q	Exits the graphical user interface.
r	Reset all the OS counters.
s	Starts or restarts the drivers and reloads the firmware.
t	Establishes the number of seconds after which a flow is considered expired (Flow Timeout).

Table 3 GUI Commands

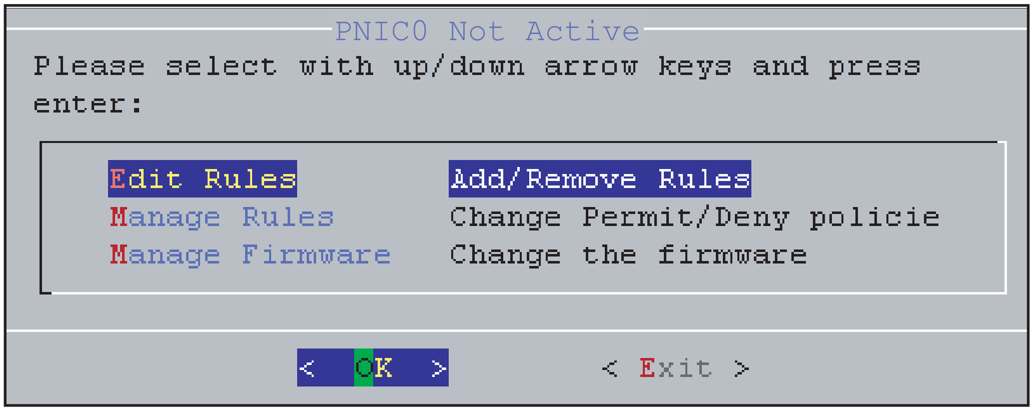
Command	Description
u	Brings the OS network interface up and enables matching. This is similar to the command **s**, but it does not load/reload the driver. It is only valid after the command **s** has been executed.
x	Toggles the direct memory access (DMA) off and on to enable or disable capturing to the host, respectively.
z	Disables the DMA and brings the interface down, in succession. This is equivalent to issuing the commands **pnic down** and **pnic off**, in succession.

Note: Commands **1**, **2**, **3**, **4**, and **5** are for engineering use only. If you enter a command **1** through **5** by mistake, enter **0** to return to the runtime statistics screen.

Managing Rules, Policies, and Firmware

Enter the **m** command from the GUI command line (see “GUI Commands” on page 26) to invoke a menu that enables you to manage dynamic rules, capture/forward policies, and firmware. Three options are available; they are shown in Figure 14 and described in Table 4.

Figure 14 Rule Management GUI



PNIC0 Not Active

Please select with up/down arrow keys and press enter:

Table 4 Managing Rules Using the GUI

Option	Description
Edit Rules	This option invokes the *vi* editor on the file *rules.custom* in the */user/local/pnic/0* directory (see “Editing Dynamic Rules with the GUI” on page 28). • You can add, delete, or modify dynamic rules for either of the processing channels (see Appendix D , on page 125 for information on *vi*). • The rules are automatically compiled and loaded into the appliance; you are prompted to confirm these actions.
Manage Rules	This option instructs the DPI on handling matching packets. • It displays a list of all the rules contained in the FPGA and the policy setting for each. • There are four policies available, and they are described in Table 5. • Rules configured to ignore a packet—that is, the policy setting is *permit* or *deny*—take precedence over rules that have a policy setting of *alert* or *divert*. Therefore, a *permit* or *deny* rule disables the capturing for all other rules that match the same packet. • To modify policy settings, see “Managing Capture/Forward Policies with the GUI” on page 29. **Note:** The **Capture** toggle is not used. Capture/forward settings can only be modified through the graphical user interface.
Manage Firmware	It displays the firmware files in */usr/local/pnic/firmware* and allows you to select one to be uploaded to the FPGA. Selecting firmware restarts and reloads the FPGA. To manage firmware, see “Selecting Firmware with the GUI” on page 30.

Table 5 describes the four possible combinations of capture/forward policies.

Table 5 Capture/Forward Policies

Policy	Capture	Forward
Permit		✓
Deny		
Alert	✓	✓
Divert	✓	

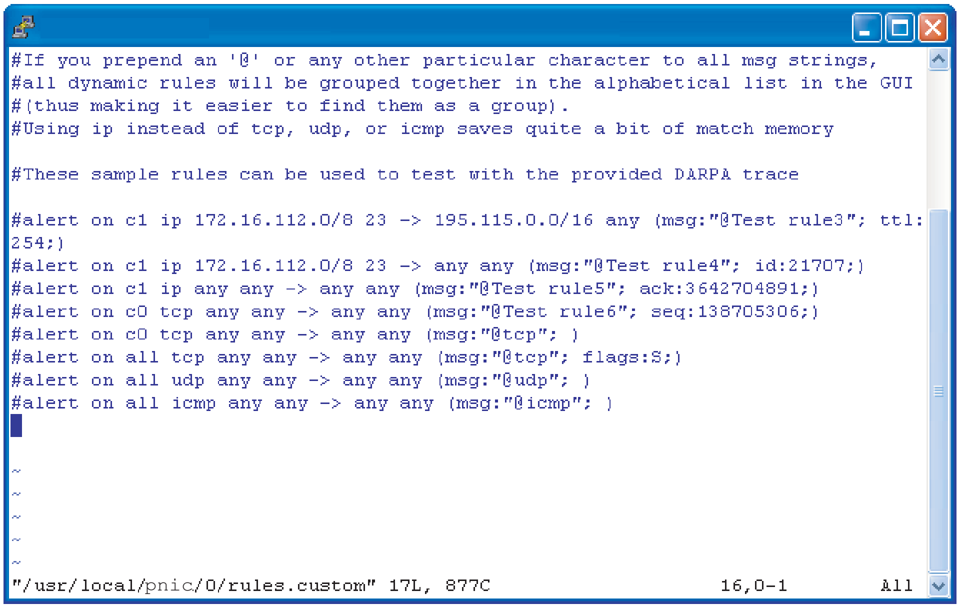
Editing Dynamic Rules with the GUI

Dynamic rules are stored in the file *rules.custom* in the */usr/local/pnic/0* directory. The GUI provides a quick way to access and modify these rules by invoking the *vi* editor on this file.

To modify dynamic rules:

Step	Task
1	Enter the **m** command from the GUI command line (see “GUI Commands” on page 26) to access the main rule management GUI (see Figure 14).
2	Select **Edit Rules** to invoke the *vi* editor (see Figure 15).
3	Add, delete, alter, or uncomment rules using *vi* commands (see Appendix D , on page 125).
4	You are prompted to confirm your changes upon exiting the editor.

Figure 15 Editing Dynamic Rules in *vi*



```
#If you prepend an '@' or any other particular character to all msg strings,
#all dynamic rules will be grouped together in the alphabetical list in the GUI
#(thus making it easier to find them as a group).
#Using ip instead of tcp, udp, or icmp saves quite a bit of match memory

#These sample rules can be used to test with the provided DARPA trace

#alert on c1 ip 172.16.112.0/8 23 -> 195.115.0.0/16 any (msg:"@Test rule3"; ttl:
254;)
#alert on c1 ip 172.16.112.0/8 23 -> any any (msg:"@Test rule4"; id:21707;)
#alert on c1 ip any any -> any any (msg:"@Test rule5"; ack:3642704891;)
#alert on c0 tcp any any -> any any (msg:"@Test rule6"; seq:138705306;)
#alert on c0 tcp any any -> any any (msg:"@tcp"; )
#alert on all tcp any any -> any any (msg:"@tcp"; flags:S;)
#alert on all udp any any -> any any (msg:"@udp"; )
#alert on all icmp any any -> any any (msg:"@icmp"; )


~
~
~
~
~
"/usr/local/pnic/0/rules.custom" 17L, 877C                    16,0-1        All
```

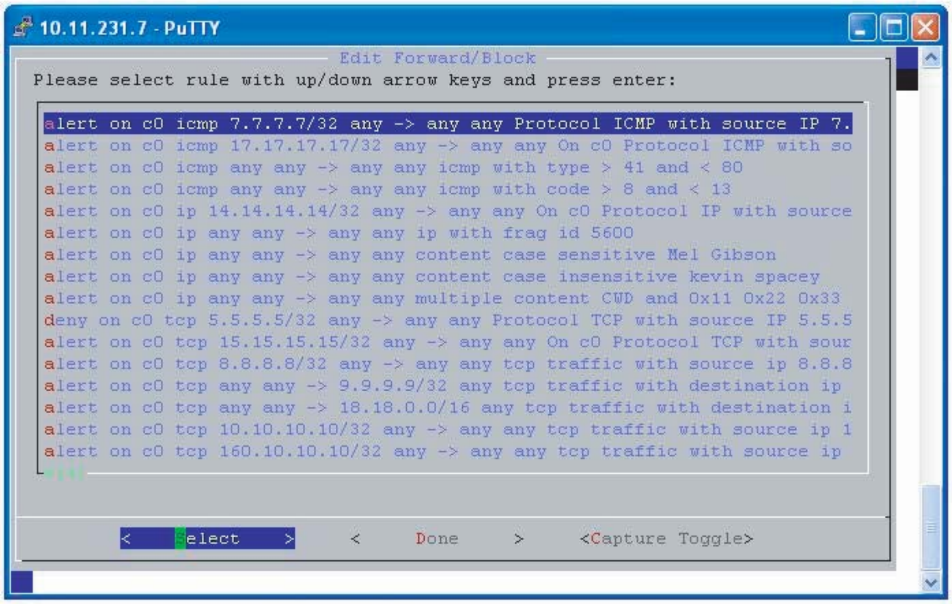
Managing Capture/Forward Policies with the GUI

Upon compiling static and dynamic rules, default capture/forward policies are assigned to each rule.

To change capture/forward policies:

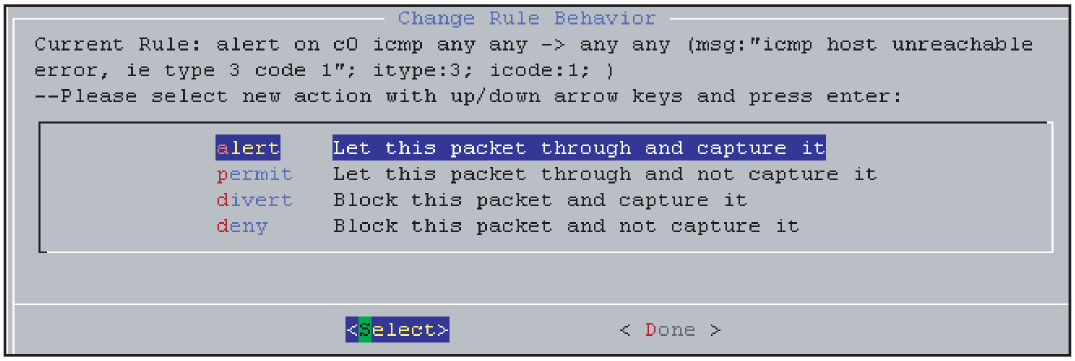
Step	Task
1	Enter the **m** command from the GUI command line (see “GUI Commands” on page 26) to access the rule management GUI (see Figure 14).
2	Select **Manage Rules** to access the policy management menu (see Figure 16).
3	Use the arrow keys to highlight a rule and the **Select** option, and press the *Enter* key.
4	Select *alert*, *permit*, *divert* or *deny*, based on the descriptions in Table 5 (also see Figure 17).
5	Exit the menu by selecting **Done**, and repeat Steps 3 through 5 for other rules, if desired.
6	Select **Done**; you are prompted to confirm your changes.

Figure 16 Managing Capture/Forward Policies GUI



```
Edit Forward/Block
Please select rule with up/down arrow keys and press enter:

alert on c0 icmp 7.7.7.7/32 any -> any any Protocol ICMP with source IP 7.
alert on c0 icmp 17.17.17.17/32 any -> any any On c0 Protocol ICMP with so
alert on c0 icmp any any -> any any icmp with type > 41 and < 80
alert on c0 icmp any any -> any any icmp with code > 8 and < 13
alert on c0 ip 14.14.14.14/32 any -> any any On c0 Protocol IP with source
alert on c0 ip any any -> any any ip with frag id 5600
alert on c0 ip any any -> any any content case sensitive Mel Gibson
alert on c0 ip any any -> any any content case insensitive kevin spacey
alert on c0 ip any any -> any any multiple content CWD and 0x11 0x22 0x33
deny on c0 tcp 5.5.5.5/32 any -> any any Protocol TCP with source IP 5.5.5
alert on c0 tcp 15.15.15.15/32 any -> any any On c0 Protocol TCP with sour
alert on c0 tcp 8.8.8.8/32 any -> any any tcp traffic with source ip 8.8.8
alert on c0 tcp any any -> 9.9.9.9/32 any tcp traffic with destination ip
alert on c0 tcp any any -> 18.18.0.0/16 any tcp traffic with destination i
alert on c0 tcp 10.10.10.10/32 any -> any any tcp traffic with source ip 1
alert on c0 tcp 160.10.10.10/32 any -> any any tcp traffic with source ip

<    Select    >     <     Done     >     <Capture Toggle>
```

Figure 17 Capture/Forward Policies GUI



```
Change Rule Behavior
Current Rule: alert on c0 icmp any any -> any any (msg:"icmp host unreachable
error, ie type 3 code 1"; itype:3; icode:1; )
--Please select new action with up/down arrow keys and press enter:

             alert    Let this packet through and capture it
             permit   Let this packet through and not capture it
             divert   Block this packet and capture it
             deny     Block this packet and not capture it

                 <Select>                    < Done >
```

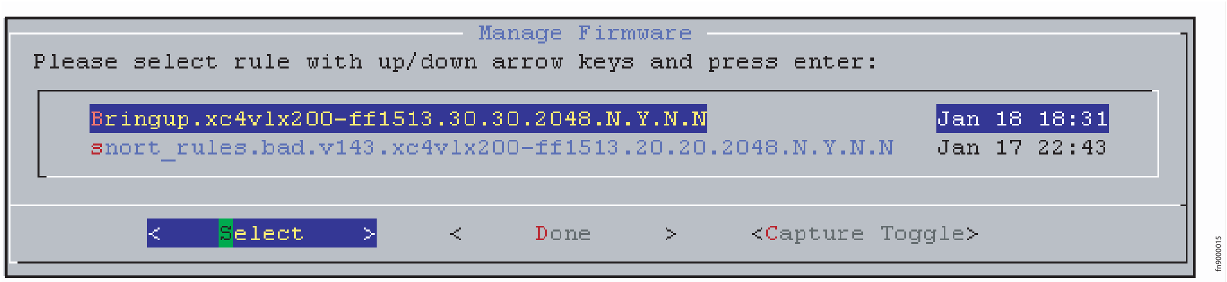
Selecting Firmware with the GUI

Firmware is a set of rules that has been transformed—using a compiler—from Snort syntax into a form suitable for uploading to the FPGA.

To select firmware:

Step	Task
1	Enter the **m** command from the GUI command line (see “GUI Commands” on page 26) to access the main rule management GUI.
2	Select **Manage Firmware** (see Figure 18).
3	Use the arrow keys to highlight the desired firmware and the **Select** option, and press the *Enter* key. See “Firmware Filename Description” on page 62 for information on identifying firmware by their filenames.
4	Confirm your selection, and exit the GUI.

Figure 18 Manage Firmware GUI



```
                         Manage Firmware
Please select rule with up/down arrow keys and press enter:

    Bringup.xc4vlx200-ff1513.30.30.2048.N.Y.N.N          Jan 18 18:31
    snort_rules.bad.v143.xc4vlx200-ff1513.20.20.2048.N.Y.N.N   Jan 17 22:43

     <    Select    >      <    Done    >     <Capture Toggle>
```

Runtime Statistics

Runtime statistics are displayed when firmware is uploaded, and traffic is flowing across the appliance. The GUI presents two views of traffic statistics. The default view shows the total statistics for Channel 0 and 1, as shown in Figure 19. Enter the command **p** to view traffic statistics for both channels separately or as a sum, as shown in Figure 20. Use the command **p** to toggle between the two views.

- The first line shows the device number, type of device, firmware ID, and version number.
- The second line shows the status of the Ethernet interface and direct memory access (DMA), and the values of Flow Timeout, Packets/Flow, and IRQ Period. These parameters can be adjusted using the GUI commands described in Table 3.

The remaining lines report the cumulative number of events and the rate of those events. A description of each line is given in Table 6.

Figure 19 Runtime Statistics for Channel 0 and 1—FPGA Loaded

```
CPU(s):   0.0% user,   0.0% system,   0.0% nice, 100.0% idle
Dev: 8002 - Type: PNIC-0 - FirmwareID: 64 - Ver:2.6 - DefaultDrop: disabled
pnic0 UP Capture=on  FlowTimeout=16 Packets/flow=0 Truncation=0 Irq period=1ms

HW Interfaces     CH0 Top                Rate/s  CH1 Top                Rate/s

Total Packets     0                       0      0                       0
   TCP Packets    0                       0      0                       0
   UDP Packets    0                       0      0                       0
   ICMP Packets   0                       0      0                       0
   Other Packets  0                       0      0                       0
Capture Packets   0                       0      0                       0
   Total Flows    0                       0      0                       0
   Delayed Pkts   0                       0      0                       0
   Stateful Pkts  0                       0      0                       0
Blocked Packets   0                       0      0                       0

OS Interface      pnic0:0                Rate/s  pnic0:1                Rate/s
Rx (Packets)      2838226                 0      2838042                 0
Rx (Bytes)        1408250941              0      1407263719              0
Rx (Bits)         2676072936              0      2668175160              0
Errors            0                       0      0                       0
Truncated (Pkts)  0                       0      0                       0
h=help z=stop m=manage_rules c=truncation t=timeout f=packets/flow x=DMA
```

Figure 20 Cumulative Runtime Statistics for Channels 0 and 1—FPGA Loaded

```
CPU(s):   0.0% user,   0.0% system,   0.0% nice, 100.0% idle
Dev: 8002 - Type: PNIC-0 - FirmwareID: 64 - Ver:2.6 - DefaultDrop: disabled
pnic0 UP Capture=on  FlowTimeout=16 Packets/flow=0 Truncation=0 Irq period=1ms

HW Interfaces     CH0 Top                Rate/s  CH1 Top                Rate/s

Total Packets     0                       0      0                       0
   TCP Packets    0                       0      0                       0
   UDP Packets    0                       0      0                       0
   ICMP Packets   0                       0      0                       0
   Other Packets  0                       0      0                       0
Capture Packets   0                       0      0                       0
   Total Flows    0                       0      0                       0
   Delayed Pkts   0                       0      0                       0
   Stateful Pkts  0                       0      0                       0
Blocked Packets   0                       0      0                       0

OS Interface                                  pnic0                Rate/s
Rx (Packets)                                  5676268               0
Rx (Bytes)                                    2815514660            0
Rx (Bits)                                     1049280800            0
Errors                                        0                     0
Truncated (Packets)                           0                     0
h=help z=stop m=manage_rules c=truncation t=timeout f=packets/flow x=DMA
```

Table 6 Runtime Statistics Description

Statistic	Description
Total Packets	Shows the number of packets received by the ports. This is a Layer 1 statistic and is independent of whether the OS interface is up or down.
TCP/UDP/ICMP/Other	Reports the type of packets received during matching. Other includes all non-IP types and all IP types other than TCP, UDP, and ICMP.
Capture Packets	Counts the total number of packets matched and captured by some policy.
Total Flows	Reports the number of new flows started according to the flow policies.
Stateful Packets	Reports the number of packets matched because of a stateful policy. The mathematical difference between this counter and the *Captured Packets* counter is the number of packets captured by stateless policies.
Blocked Packets	Reports the number of packets blocked because of some policy, except that packets blocked by default are not counted.
Rx Packets/Bytes/Bits	Tracks data received by the OS. Any difference between the values in this line and those in the *Captured Packets* line is due to buffering and/or packet loss; packet loss is due to high contention on the CPU.
Errors	Reports the number of anomalous receive conditions the driver encounters.
Truncated Packets	This feature is not supported.
Delayed Packets	Reports the number of packets that were stored in the temporary buffer in hardware.

Reloading Firmware

During firmware reloading, all packets flow regardless of capture/forward policies, as the policies cannot be enforced during system initialization. This "open" state during configuration state transition ensures that there is no interruption of service when the DPI is updated.

If the OS crashes or is halted, the device drivers are rendered inactive, but the card continues to operate independently and block/forward policies are still enforced. This behavior applies even when the device drivers are re-installed during a reboot.

Chapter 5 Web-based Management

You can manage and monitor the P-Series on the web using the Force10 Networks P-Series Node Manager.

➡ **Note:** The web-based GUI is supported on Linux only, which is the default OS, and requires software version 2.3.0.0 or newer.

Launching the P-Series Node Manager

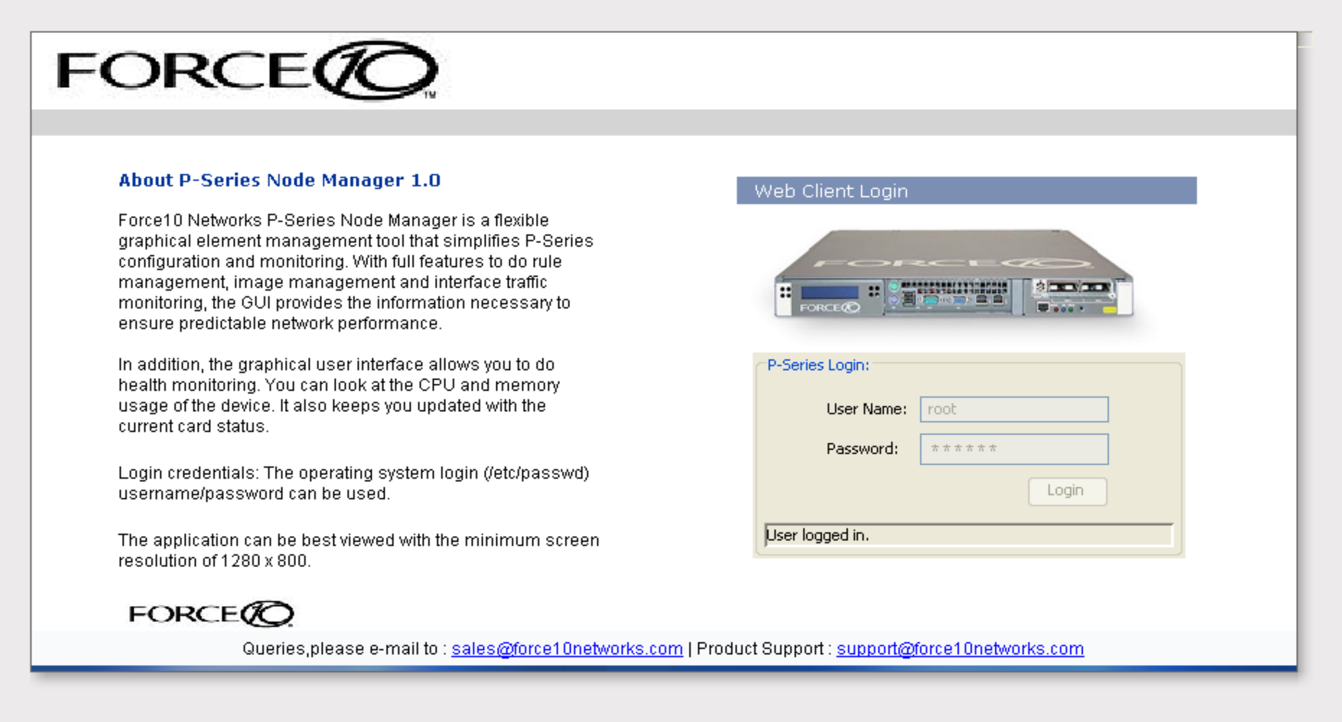
➡ **Note:** The Web-based GUI is best viewed with a minimum screen resolution of 1280x800. You must also have Java Run Time Environment (JRE) installed with the “Use JRE X.Y.Z for <applet>” option enabled under Tool --> Internet Options --> Advanced tab when using either Internet Explorer 6 or 7.

To launch the P-Series Node Manager:

Step	Task
1	Enter the command **pnic web-gui-start** to enable the secure HTTP service on the P-Series (see Appendix A , on page 79).
2	Lauch the P-Series Node Manager in a web brower by entering **https://*ipaddress*** in the address bar, as shown in Figure 21.
3	Login using the username and password configured on your P-Series appliance.

Note: Stop the secure HTTP service using the command **pnic web-gui-stop** (see Appendix A , on page 79).

Figure 21 Lauching the P-Series Node Manager



About P-Series Node Manager 1.0

Force10 Networks P-Series Node Manager is a flexible graphical element management tool that simplifies P-Series configuration and monitoring. With full features to do rule management, image management and interface traffic monitoring, the GUI provides the information necessary to ensure predictable network performance.

In addition, the graphical user interface allows you to do health monitoring. You can look at the CPU and memory usage of the device. It also keeps you updated with the current card status.

Login credentials: The operating system login (/etc/passwd) username/password can be used.

The application can be best viewed with the minimum screen resolution of 1280 x 800.

Web Client Login

P-Series Login:

User Name: root

Password: ******

Login

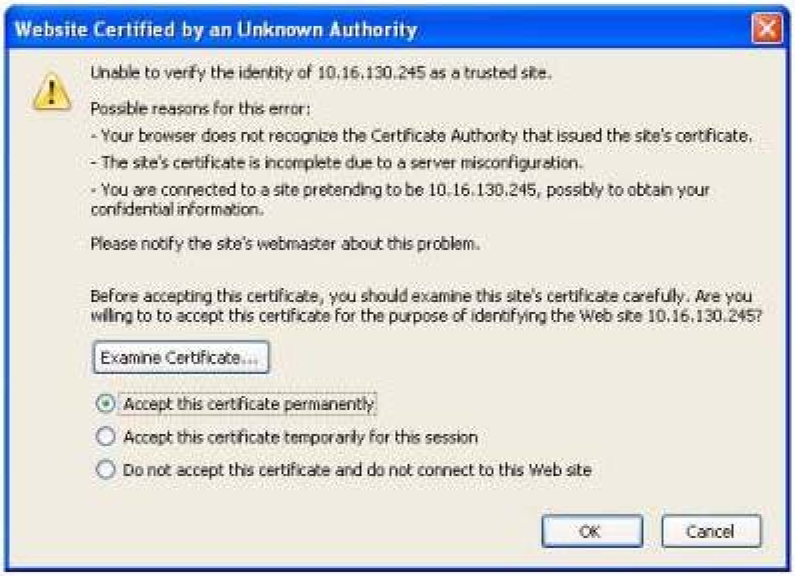
User logged in.

Queries,please e-mail to : sales@force10networks.com | Product Support : support@force10networks.com

Web-browser Security Certificates

The P-Series Node Manager client and the server communicate via HTTPs. All transactions are encrypted, and thus protected, by the SSL protocol. The SSL certificate is a self-signed certificate that is not signed by a trusted Certificate Authority (CA). While trying to launch the P-Series Node Manager, your web browser might display an alert indicating that the security certificate was not issued by trusted CA or a similar warning (Figure 22). You are safe to use the application without security risks.

Figure 22 Web-browser Security Certificate Alert



Website Certified by an Unknown Authority

Unable to verify the identity of 10.16.130.245 as a trusted site.

Possible reasons for this error:

- Your browser does not recognize the Certificate Authority that issued the site's certificate.
- The site's certificate is incomplete due to a server misconfiguration.
- You are connected to a site pretending to be 10.16.130.245, possibly to obtain your confidential information.

Please notify the site's webmaster about this problem.

Before accepting this certificate, you should examine this site's certificate carefully. Are you willing to to accept this certificate for the purpose of identifying the Web site 10.16.130.245?

Examine Certificate...

● Accept this certificate permanently

○ Accept this certificate temporarily for this session

○ Do not accept this certificate and do not connect to this Web site

OK Cancel

Managing the P-Series using Node Manager

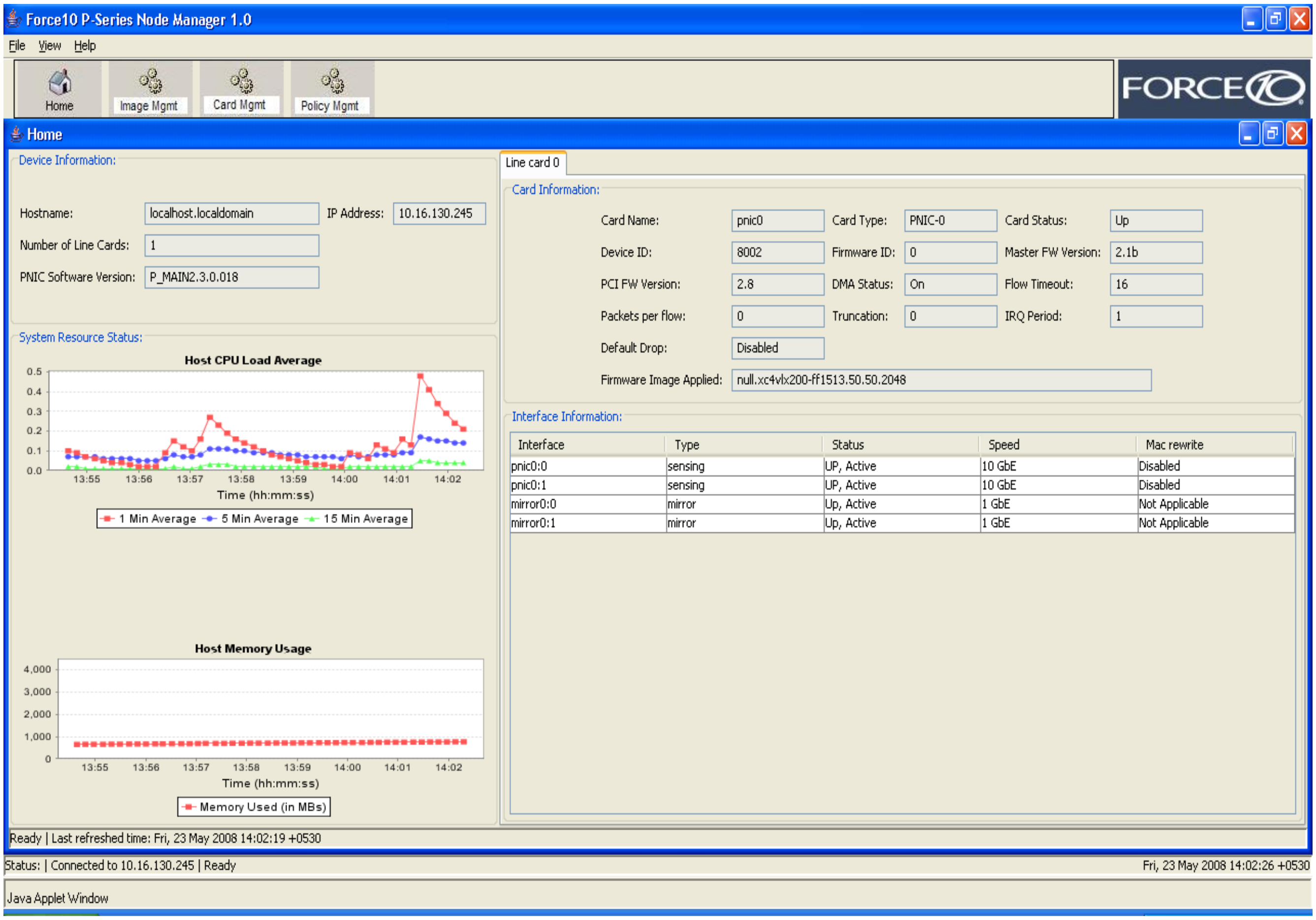
P-Series Node Manager has four major management capabilities:

- Monitoring System Performance on page 38
- Managing Firmware Images on page 39
- Managing the Network Interface Card on page 39
- Managing Policies on page 41

Monitoring System Performance

Monitor system performance from the Home panel (Figure 23). The Home panel is displayed after logging into Node Manager. It displays basic system information, card, interface, and resource information, as well as CPU and memory usage over time.

Figure 23 P-Series Node Manager: Home Panel



Device Information:

Hostname: localhost.localdomain IP Address: 10.16.130.245

Number of Line Cards: 1

PNIC Software Version: P_MAIN2.3.0.018

Card Information:

Card Name:	pnic0	Card Type:	PNIC-0	Card Status:	Up
Device ID:	8002	Firmware ID:	0	Master FW Version:	2.1b
PCI FW Version:	2.8	DMA Status:	On	Flow Timeout:	16
Packets per flow:	0	Truncation:	0	IRQ Period:	1
Default Drop:	Disabled				

Firmware Image Applied: null.xc4vlx200-ff1513.50.50.2048

Interface Information:

Interface	Type	Status	Speed	Mac rewrite
pnic0:0	sensing	UP, Active	10 GbE	Disabled
pnic0:1	sensing	UP, Active	10 GbE	Disabled
mirror0:0	mirror	Up, Active	1 GbE	Not Applicable
mirror0:1	mirror	Up, Active	1 GbE	Not Applicable

Ready | Last refreshed time: Fri, 23 May 2008 14:02:19 +0530

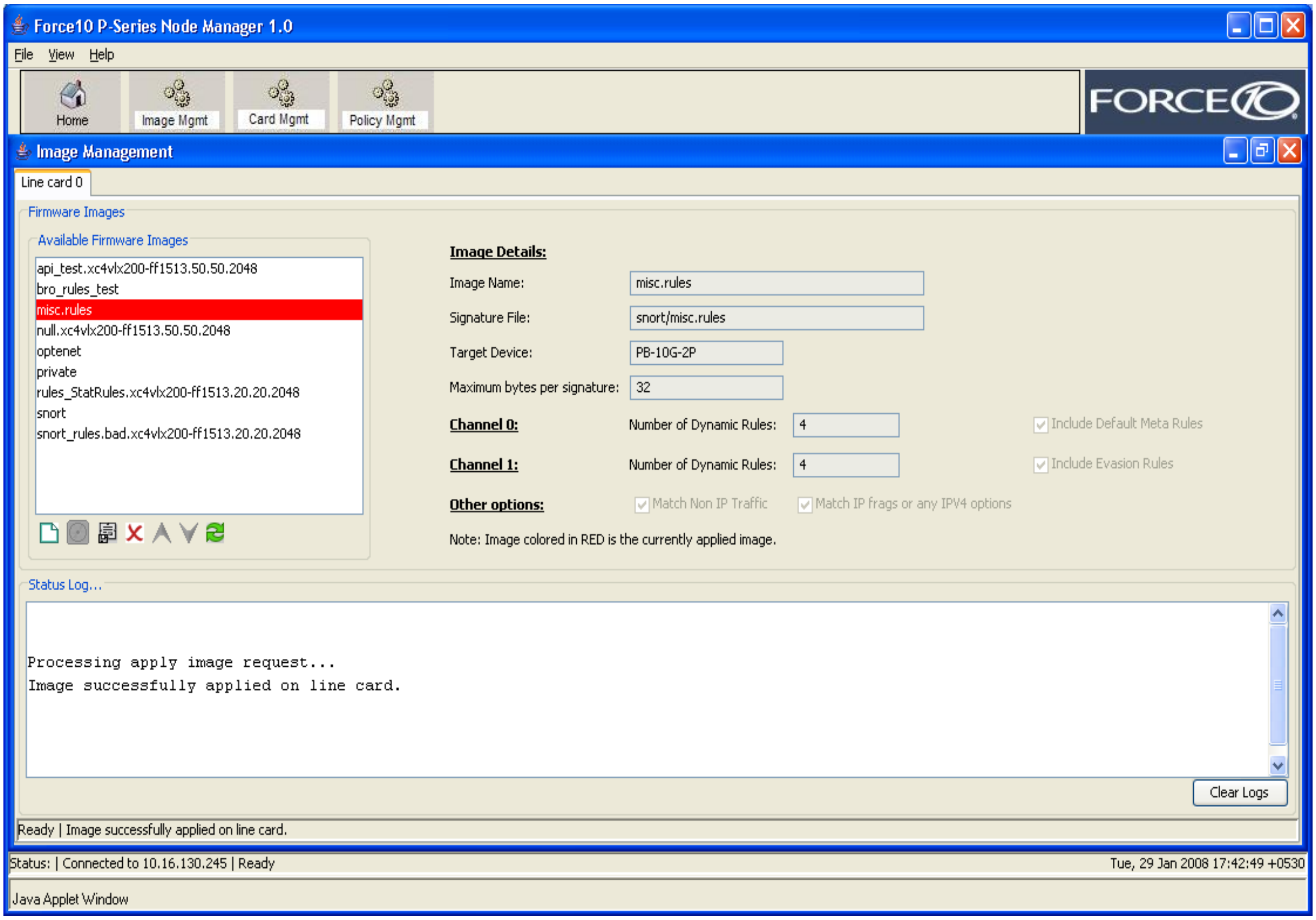
Status: | Connected to 10.16.130.245 | Ready Fri, 23 May 2008 14:02:26 +0530

Java Applet Window

Managing Firmware Images

Manage the software image from the Image Management panel (Figure 24). The Image Management panel provides options for compiling and deleting an image. It displays a list of available images along with the currently applied image and its details.

Figure 24 P-Series Node Manager: Image Managment Panel



api_test.xc4vlx200-ff1513.50.50.2048
bro_rules_test
misc.rules
null.xc4vlx200-ff1513.50.50.2048
optenet
private
rules_StatRules.xc4vlx200-ff1513.20.20.2048
snort
snort_rules.bad.xc4vlx200-ff1513.20.20.2048

Image Details:

Image Name: misc.rules

Signature File: snort/misc.rules

Target Device: PB-10G-2P

Maximum bytes per signature: 32

Channel 0: Number of Dynamic Rules: 4 ☑ Include Default Meta Rules

Channel 1: Number of Dynamic Rules: 4 ☑ Include Evasion Rules

Other options: ☑ Match Non IP Traffic ☑ Match IP frags or any IPV4 options

Note: Image colored in RED is the currently applied image.

Status Log...

```
Processing apply image request...
Image successfully applied on line card.
```

Ready | Image successfully applied on line card.

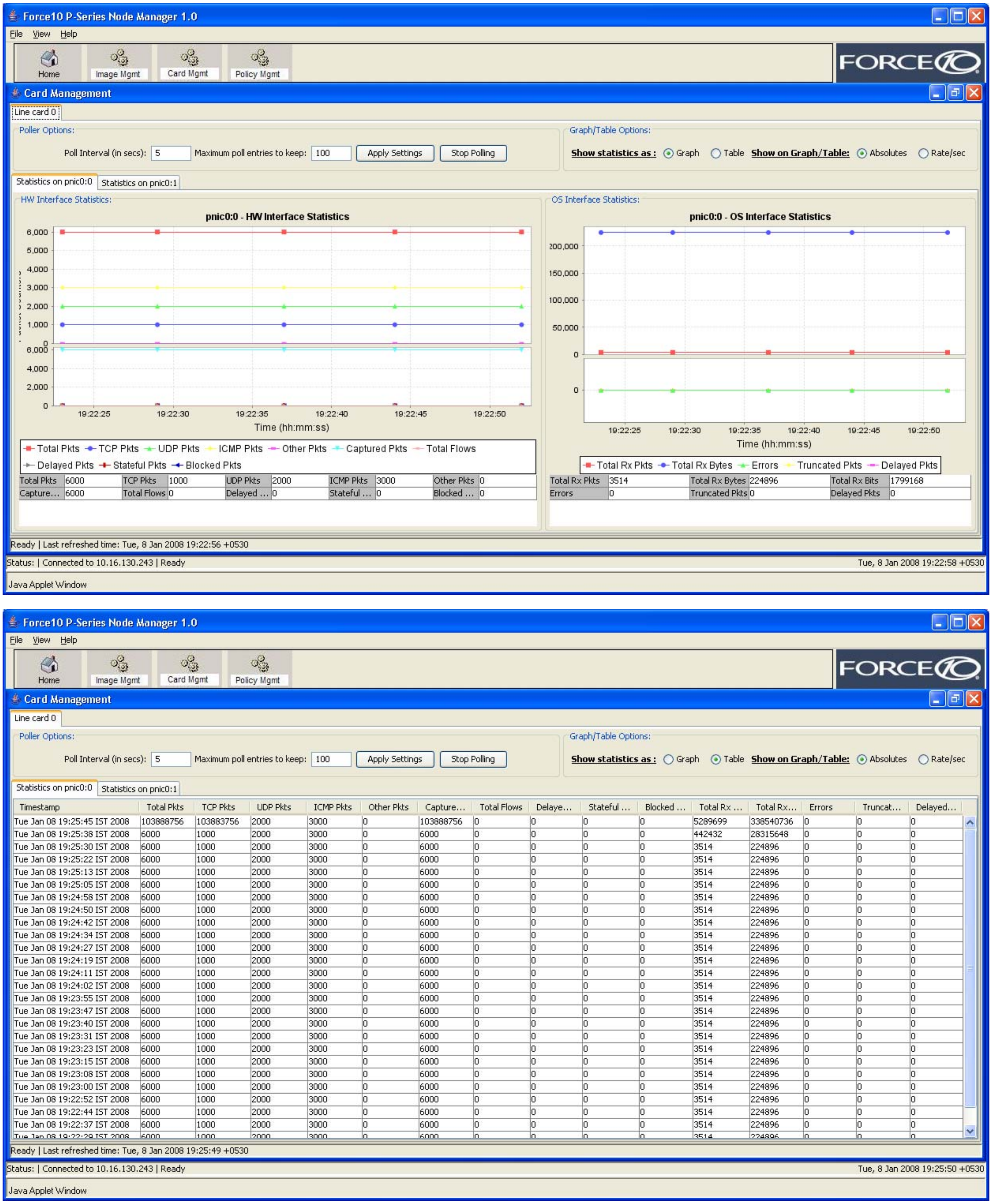
Status: | Connected to 10.16.130.245 | Ready Tue, 29 Jan 2008 17:42:49 +0530

Java Applet Window

Managing the Network Interface Card

Manage the network interface card from the Card Management panel. The Card Management panel displays hardware and software counters for Channel 0 (pnic 0:0) and Channel 1 (pnic 0:1). Counters are displayed in absolute value and in graphical or tabular format, as shown in Figure 25.

Figure 25 P-Series Node Manager: Card Management Panel



Poller Options:

Poll Interval (in secs): 5 Maximum poll entries to keep: 100 Apply Settings Stop Polling

Graph/Table Options:

Show statistics as : ● Graph ○ Table **Show on Graph/Table:** ● Absolutes ○ Rate/sec

Statistics on pnic0:0 | Statistics on pnic0:1

HW Interface Statistics:

Total Pkts	6000	TCP Pkts	1000	UDP Pkts	2000	ICMP Pkts	3000	Other Pkts	0
Capture...	6000	Total Flows	0	Delayed ...	0	Stateful ...	0	Blocked ...	0

OS Interface Statistics:

Total Rx Pkts	3514	Total Rx Bytes	224896	Total Rx Bits	1799168
Errors	0	Truncated Pkts	0	Delayed Pkts	0

Ready | Last refreshed time: Tue, 8 Jan 2008 19:22:56 +0530

Status: | Connected to 10.16.130.243 | Ready Tue, 8 Jan 2008 19:22:58 +0530

Java Applet Window

Poller Options:

Poll Interval (in secs): 5 Maximum poll entries to keep: 100 Apply Settings Stop Polling

Graph/Table Options:

Show statistics as : ○ Graph ● Table **Show on Graph/Table:** ● Absolutes ○ Rate/sec

Statistics on pnic0:0 | Statistics on pnic0:1

Timestamp	Total Pkts	TCP Pkts	UDP Pkts	ICMP Pkts	Other Pkts	Capture...	Total Flows	Delaye...	Stateful ...	Blocked ...	Total Rx ...	Total Rx...	Errors	Truncat...	Delayed...
Tue Jan 08 19:25:45 IST 2008	103888756	103883756	2000	3000	0	103888756	0	0	0	0	5289699	338540736	0	0	0
Tue Jan 08 19:25:38 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	442432	28315648	0	0	0
Tue Jan 08 19:25:30 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:25:22 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:25:13 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:25:05 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:58 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:50 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:42 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:34 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:27 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:19 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:11 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:24:02 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:55 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:47 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:40 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:31 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:23 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:15 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:08 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:23:00 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:22:52 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:22:44 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:22:37 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0
Tue Jan 08 19:22:29 IST 2008	6000	1000	2000	3000	0	6000	0	0	0	0	3514	224896	0	0	0

Ready | Last refreshed time: Tue, 8 Jan 2008 19:25:49 +0530

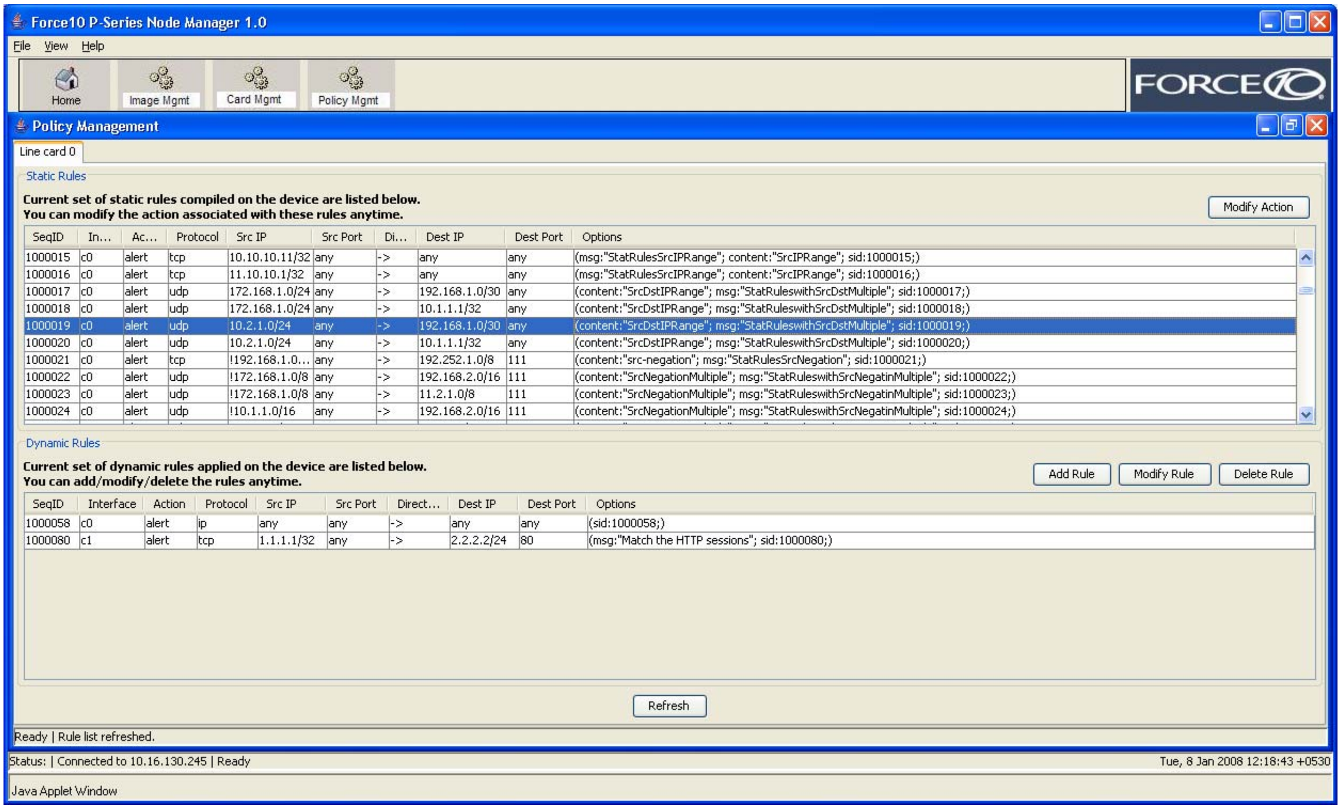
Status: | Connected to 10.16.130.243 | Ready Tue, 8 Jan 2008 19:25:50 +0530

Java Applet Window

Managing Policies

Manage policies from the Policy Management panel (Figure 26). The Policy Management panel provides you with a list of available static and dynamic rules available for the currently running image. It also has the provision for adding, modifying, and deleting dynamic rules.

Figure 26 P-Series Node Manager: Policy Managment Panel



Current set of static rules compiled on the device are listed below.
You can modify the action associated with these rules anytime.

SeqID	In...	Ac...	Protocol	Src IP	Src Port	Di...	Dest IP	Dest Port	Options
1000015	c0	alert	tcp	10.10.10.11/32	any	->	any	any	(msg:"StatRulesSrcIPRange"; content:"SrcIPRange"; sid:1000015;)
1000016	c0	alert	tcp	11.10.10.1/32	any	->	any	any	(msg:"StatRulesSrcIPRange"; content:"SrcIPRange"; sid:1000016;)
1000017	c0	alert	udp	172.168.1.0/24	any	->	192.168.1.0/30	any	(content:"SrcDstIPRange"; msg:"StatRuleswithSrcDstMultiple"; sid:1000017;)
1000018	c0	alert	udp	172.168.1.0/24	any	->	10.1.1.1/32	any	(content:"SrcDstIPRange"; msg:"StatRuleswithSrcDstMultiple"; sid:1000018;)
1000019	c0	alert	udp	10.2.1.0/24	any	->	192.168.1.0/30	any	(content:"SrcDstIPRange"; msg:"StatRuleswithSrcDstMultiple"; sid:1000019;)
1000020	c0	alert	udp	10.2.1.0/24	any	->	10.1.1.1/32	any	(content:"SrcDstIPRange"; msg:"StatRuleswithSrcDstMultiple"; sid:1000020;)
1000021	c0	alert	tcp	!192.168.1.0...	any	->	192.252.1.0/8	111	(content:"src-negation"; msg:"StatRulesSrcNegation"; sid:1000021;)
1000022	c0	alert	udp	!172.168.1.0/8	any	->	192.168.2.0/16	111	(content:"SrcNegationMultiple"; msg:"StatRuleswithSrcNegatinMultiple"; sid:1000022;)
1000023	c0	alert	udp	!172.168.1.0/8	any	->	11.2.1.0/8	111	(content:"SrcNegationMultiple"; msg:"StatRuleswithSrcNegatinMultiple"; sid:1000023;)
1000024	c0	alert	udp	!10.1.1.0/16	any	->	192.168.2.0/16	111	(content:"SrcNegationMultiple"; msg:"StatRuleswithSrcNegatinMultiple"; sid:1000024;)

Current set of dynamic rules applied on the device are listed below.
You can add/modify/delete the rules anytime.

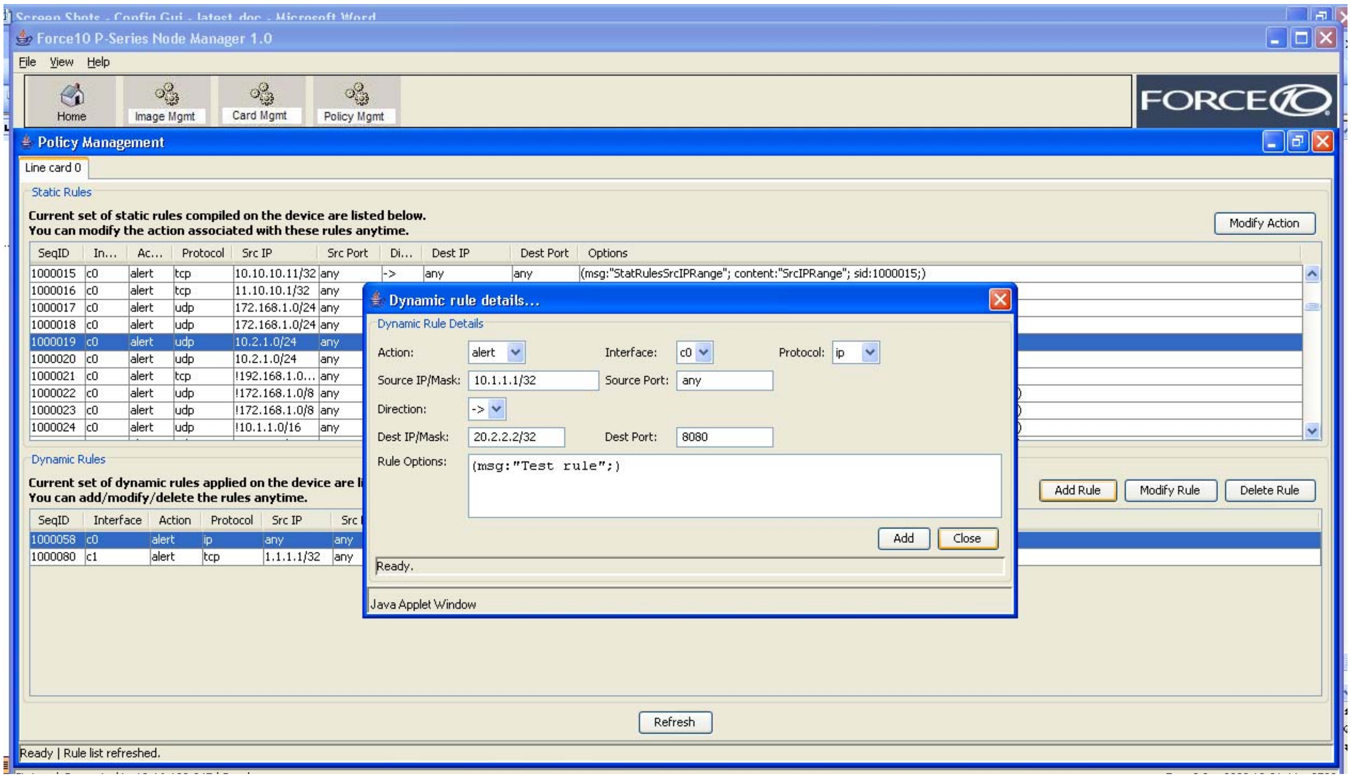
SeqID	Interface	Action	Protocol	Src IP	Src Port	Direct...	Dest IP	Dest Port	Options
1000058	c0	alert	ip	any	any	->	any	any	(sid:1000058;)
1000080	c1	alert	tcp	1.1.1.1/32	any	->	2.2.2.2/24	80	(msg:"Match the HTTP sessions"; sid:1000080;)

Ready | Rule list refreshed.

Status: | Connected to 10.16.130.245 | Ready

Tue, 8 Jan 2008 12:18:43 +0530

Java Applet Window



Action: alert

Interface: c0

Protocol: ip

Source IP/Mask: 10.1.1.1/32

Source Port: any

Direction: ->

Dest IP/Mask: 20.2.2.2/32

Dest Port: 8080

Rule Options: (msg:"Test rule";)

Ready.

Java Applet Window

Current set of static rules compiled on the device are listed below.
You can modify the action associated with these rules anytime.

SeqID	In...	Ac...	Protocol	Src IP	Src Port
1000015	c0	alert	tcp	10.10.10.11/32	any
1000016	c0	alert	tcp	11.10.10.1/32	any
1000017	c0	alert	udp	172.168.1.0/24	any
1000018	c0	alert	udp	172.168.1.0/24	any
1000019	c0	alert	udp	10.2.1.0/24	any
1000020	c0	alert	udp	10.2.1.0/24	any
1000021	c0	alert	tcp	!192.168.1.0...	any
1000022	c0	alert	udp	!172.168.1.0/8	any
1000023	c0	alert	udp	!172.168.1.0/8	any
1000024	c0	alert	udp	!10.1.1.0/16	any

Dynamic Rules

Current set of dynamic rules applied on the device are li
You can add/modify/delete the rules anytime.

SeqID	Interface	Action	Protocol	Src IP	Src
1000058	c0	alert	ip	any	any
1000080	c1	alert	tcp	1.1.1.1/32	any

Add Rule Modify Rule Delete Rule

Refresh

Ready | Rule list refreshed.

Chapter 6 Network Security Monitoring

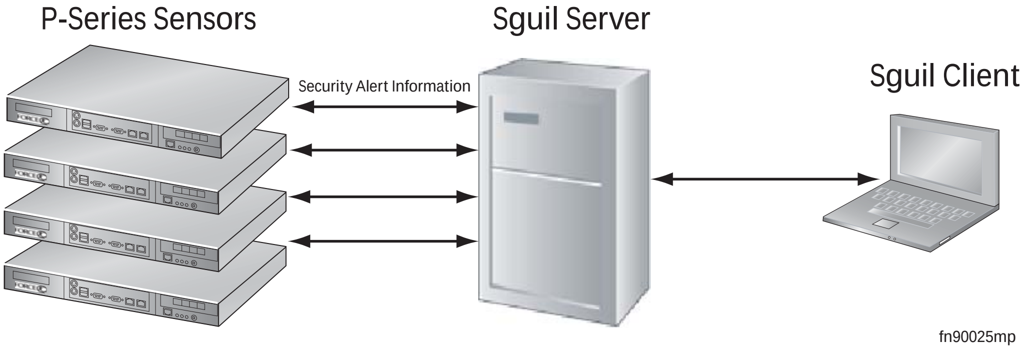
A key aspect of network security deployment is the ability to monitor the network for security events, analyze them, and perform counter measures. To that end, the P-Series supports Sguil, an open source network security monitoring and reporting system that provides the ability to:

- collect, monitor, and correlate security events/alerts in the network
- analyze security events based on context
- categorize and escalate events for intrusion response decisions

The Sguil solution consists of the following components (Figure 27):

- **Sensors**—Sensors are the systems actually monitoring network traffic and collecting data. Sensors perform packet captures of network traffic in addition to running Snort in alert mode.
- **Database**—The database holds the alert and session data that the sensors collect.
- **Client**—The client is the interface to the Sguil server.
- **Server**—The Sguil server maintains connections to the sensors, clients, and database.

Figure 27 Sguil Architecture



Installing the Sguil System

To employ Sguil you must:

1. Install the sensor. See page 44.
2. Install the server. See page 44.
3. Install the client. See page 45.

➡ **Note:** You can download the server and client Sguil components directly from the Sguil website at http://sguil.sourceforge.net/index.html. The solution uses a number of components which must be installed. For your convenience, a simplified install package is provided on the Force10 Networks support website; please see the instructions in the remainder of this chapter.

Installing the Sguil Sensor

P-Series appliances running version 2.3.0.0 or newer are already capable of operating as a Sguil sensor.

Installing the Sguil Server

The Sguil server package installs the Mysql server and Sguild server packages.

Hardware and Software Requirements

Force10 recommends using a server that has at least 2 GB of RAM, a 3.0 GHz processor, and 150 GB hard disk with a RAID5 array for speed and reliability.

Sguil runs on a variety of *BSD and Linux-based systems. Force10 has tested compatibility with and recommends using:

- CentOs 5 64 bit Linux version 2.6.18-8.1.14.el5
- CentOs 5 32 bit Linux version 2.6.18-8.1.14.el5, or
- FreeBSD-6.2-<release>

➡ **Note:** Red Hat Enterprise Linux (RHEL) might also be compatible but has not been tested.

To install the server:

Step	Task	Command
1	Copy *sguil-server-<version>.tar.gz* to the server in which it will be installed.	
2	From the directory where the server package is stored, untar the Sguil server package.	**tar -zxvf sguil-server-<*version*>.tar.gz**
3	Change to Bash shell.	**bash**

Step	Task	Command
4	Source the server configuration file. The default parameters in this file may be changed.	**source Configure-Inputs.sh**
5	Compile and build the Sguil server package. Use the logging option to collect debugging information during compilation and redirect standard output and errors to a log file.	**gmake [> build.log 2>&1]**
6	Install the Sguil server package.	**gmake install**
7	(OPTIONAL) Set the debug flag to 1 in *sguild.conf* before executing Startserver.sh to display Sguil server debug messages	

Uninstalling the Sguil Server

To uninstall the server:

Step	Task	Command
1	Stop the Sguil and MySQL servers, if they are running.	
2	From the directory in which the sever package was installed, source the Sguil server configuration file.	**source Configure-Inputs.sh**
3	Uninstall the Sguil server. Use the logging option to collect debugging information during uninstallation and redirect standard output and errors to a log file.	**gmake uninstall** [**> uninstall.log 2>&1**]

Installing the Sguil Client

You must have the following software installed in your PC before installing the Sguil client:

- ActiveTcl, Force10 recommends ActiveTcl8.4.14 which includes Wish
- WinZip
- Wireshark
- Wish
- Download the OpenSSL TCL extension TLS package to the client and extract the contents to the *lib* directory of the TCL installation. Typically the TCL installation directory is *c:\program files\tcl*.

To install the client:

Step	Task
1	Copy *sguil-client-<version>.tar.gz* to the PC on which it will be installed.
2	Extract the tar file.

Step	Task
3	Configure the following parameters in the file *sguil.conf*: • Enable (1) or disable (0) the debug option • Set the browser path. • Set the Wireshark application path. • Set the TLS library path, as shown in Figure 28. • Set priority levels of the alert window.

Figure 28 Setting the TLS Library Path

```
# PATH to tls lib if needed (tcl can usually find this by default)
#set TLS_PATH /usr/lib/tls1.4/libtls1.4.so
# win32 example
set TLS_PATH "c:/progra~1/Tcl/lib/tls1.4.1/tls14.dll"
```

Installation Files

Table 7 lists the files and directories created during installation that are relevant to running the Sguil system.

Table 7 Sguil Files and Directories

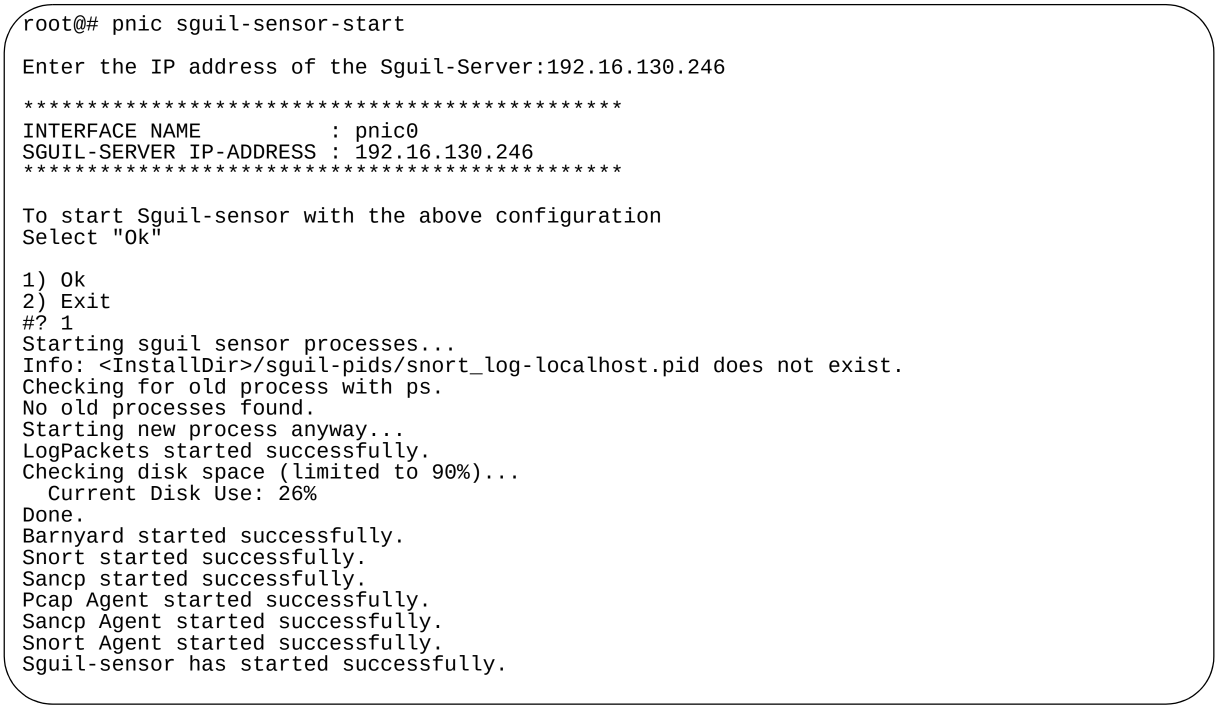
File	Location
Sensor	
sensor installation directory	/usr/local/pnic-mgmt-lib/sguil-sensor
sensor configuration files	<install_dir>/nsm/sguil/etc
snort.conf	<install_dir>/nsm/sguil/etc/
log files	<install_dir>/nsm/sguil/logs
rules files	<install_dir>/nsm/sguil/rules
Snort logs	/var/log/Snort
Packet logs	/var/log/Sensor/LogPackets
Server	
server installation directory	/usr/local/sguil-server
sguild.conf	<install_dir>/nsm/sguil/etc
log files	<install_dir>/nsm/sguil/logs

Running the Sguil System

Running the Sguil Sensor

Start the Sguil sensor using the command **pnic sguil-sensor-start**. Specify the IP address of the Sguil server, and confirm the action, as shown in Figure 29.

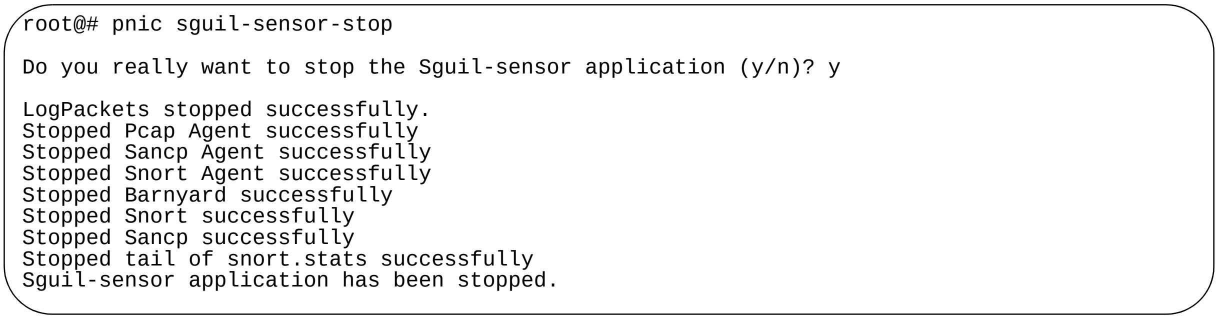
Figure 29 Starting the Sguil Sensor



```
root@# pnic sguil-sensor-start

Enter the IP address of the Sguil-Server:192.16.130.246

***********************************************
INTERFACE NAME          : pnic0
SGUIL-SERVER IP-ADDRESS : 192.16.130.246
***********************************************

To start Sguil-sensor with the above configuration
Select "Ok"

1) Ok
2) Exit
#? 1
Starting sguil sensor processes...
Info: <InstallDir>/sguil-pids/snort_log-localhost.pid does not exist.
Checking for old process with ps.
No old processes found.
Starting new process anyway...
LogPackets started successfully.
Checking disk space (limited to 90%)...
  Current Disk Use: 26%
Done.
Barnyard started successfully.
Snort started successfully.
Sancp started successfully.
Pcap Agent started successfully.
Sancp Agent started successfully.
Snort Agent started successfully.
Sguil-sensor has started successfully.
```

Stop the Sguil sensor using the command **pnic sguil-sensor-stop**, as shown in Figure 30.

Figure 30 Stopping the Sguil Sensor



```
root@# pnic sguil-sensor-stop

Do you really want to stop the Sguil-sensor application (y/n)? y

LogPackets stopped successfully.
Stopped Pcap Agent successfully
Stopped Sancp Agent successfully
Stopped Snort Agent successfully
Stopped Barnyard successfully
Stopped Snort successfully
Stopped Sancp successfully
Stopped tail of snort.stats successfully
Sguil-sensor application has been stopped.
```

Writing New Rules

- All rules files are stored in the installation sub-directory *../nsm/sguil/rules*.

- The rule file you are using should be mentioned in *snort.conf* file. A sample rule file under rules directory is already added and commented in *snort.conf*.
- Log files are stored in the installation sub-directory *.../nsm/sguil/logs*.
- When adding new rules to the file *sample.rules*, uncomment the line, “include sample.rules”in the file *snort.conf*.
- Snort rule syntax is different from P-Series rule syntax. For example, the following rule is invalid for Snort, but valid for the P-Series: *alert on c1 tcp any any ->any any (msg:”tcp”; sid:100000001; rev:1;)*. See Chapter 9, Writing Rules, on page 63.
  - The SID rule option is mandatory for Snort rules.
  - Do not specify channel information in Snort rules as it is already specified in P-Series rules and will yeild a syntax error.

Running the Sguil Server

Scripts are used to perform management tasks such as starting and stopping the server and adding and deleting users. Run scripts from the *bin* sub-directory of the installation directory.

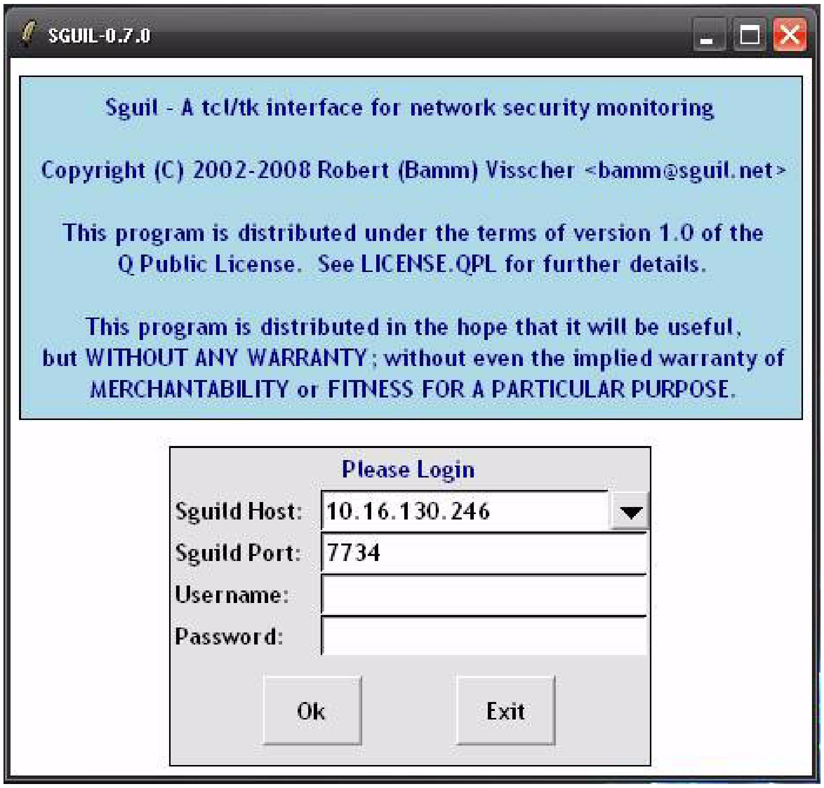
Task	Script
Start the server. When the Sguild server is started for the first time, you are prompted to add a new user.	**./StartMysqlserver.sh** **./Startserver.sh**
Stop the server.	**./Shutdownserver.sh** **./ShutdownMysqlserver.sh**
Add a new user. You are prompted for a new username and password.	**./ManageSguilserverUser.sh add**
Delete a user. You are prompted for your username and Squil user to be deleted.	**./ManageSguilserverUser.sh delete**

Running the Sguil Client

To run the Sguil Client:

Step	Task
1	Open *sguil.tk* using the Wish application. A window appears, as shown in Figure 31.
2	Specify the IP address of the Sguil server, and your username and password.
3	Select the sensors to monitor (click “Select All” to monitor all sensors), and click “Start SGUIL” (Figure 32).

Figure 31 Running the Sguil Client



Sguil - A tcl/tk interface for network security monitoring

Copyright (C) 2002-2008 Robert (Bamm) Visscher <bamm@sguil.net>

This program is distributed under the terms of version 1.0 of the
Q Public License. See LICENSE.QPL for further details.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Please Login

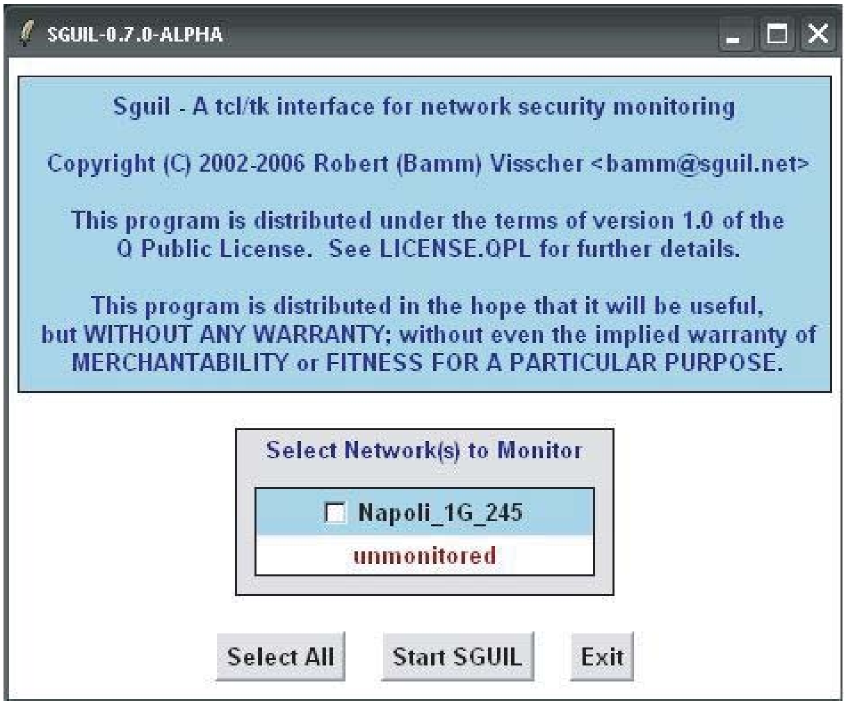
Sguild Host: 10.16.130.246

Sguild Port: 7734

Username:

Password:

Figure 32 Selecting the Sensor to Monitor



Sguil - A tcl/tk interface for network security monitoring

Copyright (C) 2002-2006 Robert (Bamm) Visscher <bamm@sguil.net>

This program is distributed under the terms of version 1.0 of the Q Public License. See LICENSE.QPL for further details.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Select Network(s) to Monitor

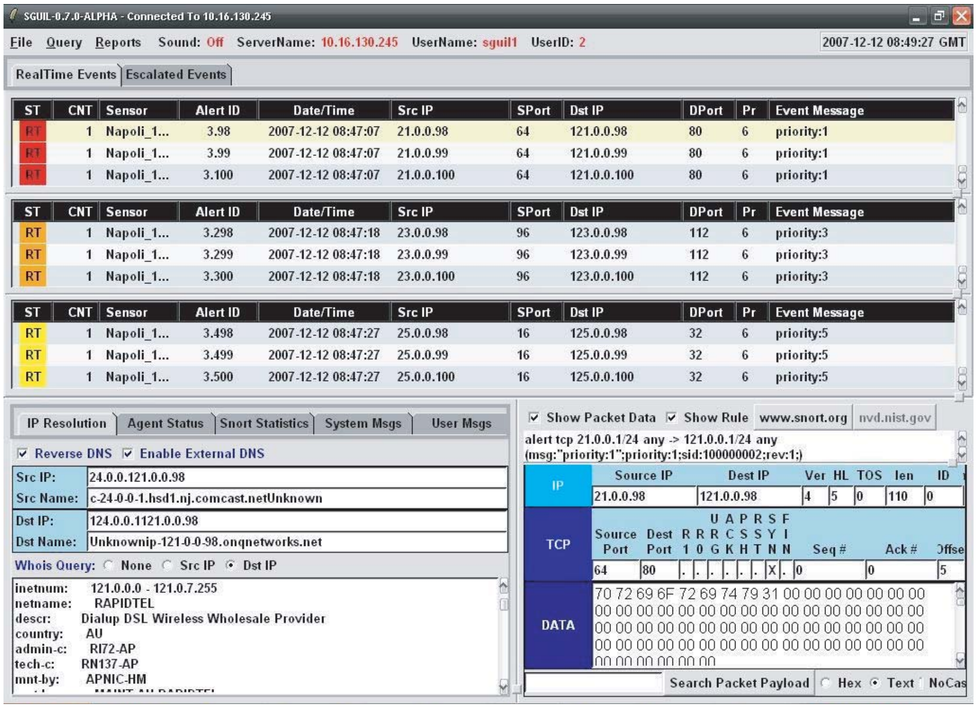
□ Napoli_1G_245

unmonitored

fn90027mp

When the Sguil client starts and the client is properly connected to the Sguil server, the window in Figure 33 appears.

Figure 33 Accepting Events from the Sensor



Sound: Off ServerName: 10.16.130.245 UserName: sguil1 UserID: 2 2007-12-12 08:49:27 GMT

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	Napoli_1...	3.98	2007-12-12 08:47:07	21.0.0.98	64	121.0.0.98	80	6	priority:1
RT	1	Napoli_1...	3.99	2007-12-12 08:47:07	21.0.0.99	64	121.0.0.99	80	6	priority:1
RT	1	Napoli_1...	3.100	2007-12-12 08:47:07	21.0.0.100	64	121.0.0.100	80	6	priority:1

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	Napoli_1...	3.298	2007-12-12 08:47:18	23.0.0.98	96	123.0.0.98	112	6	priority:3
RT	1	Napoli_1...	3.299	2007-12-12 08:47:18	23.0.0.99	96	123.0.0.99	112	6	priority:3
RT	1	Napoli_1...	3.300	2007-12-12 08:47:18	23.0.0.100	96	123.0.0.100	112	6	priority:3

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	Napoli_1...	3.498	2007-12-12 08:47:27	25.0.0.98	16	125.0.0.98	32	6	priority:5
RT	1	Napoli_1...	3.499	2007-12-12 08:47:27	25.0.0.99	16	125.0.0.99	32	6	priority:5
RT	1	Napoli_1...	3.500	2007-12-12 08:47:27	25.0.0.100	16	125.0.0.100	32	6	priority:5

☑ Reverse DNS ☑ Enable External DNS

Src IP: 24.0.0.121.0.0.98
Src Name: c-24-0-0-1.hsd1.nj.comcast.netUnknown
Dst IP: 124.0.0.1121.0.0.98
Dst Name: Unknownip-121-0-0-98.onqnetworks.net

Whois Query: ○ None ○ Src IP ● Dst IP

```
inetnum:      121.0.0.0 - 121.0.7.255
netname:      RAPIDTEL
descr:       Dialup DSL Wireless Wholesale Provider
country:     AU
admin-c:      RI72-AP
tech-c:      RN137-AP
mnt-by:      APNIC-HM
```

☑ Show Packet Data ☑ Show Rule

```
alert tcp 21.0.0.1/24 any -> 121.0.0.1/24 any
(msg:"priority:1";priority:1;sid:100000002;rev:1;)
```

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID
	21.0.0.98	121.0.0.98	4	5	0	110	0

TCP	Source Port	Dest Port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	Seq #	Ack #	Offse
	64	80	.	.	.	.	.	.	X	.	0	0	5

DATA

```
70 72 69 6F 72 69 74 79 31 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
```

○ Hex ● Text NoCas

fn90028mp

Chapter 7 Command Line Interface

The command line interface (CLI) is an alternative to the GUI for managing the appliance. A script called *pnic* is used to perform the same management functions as the GUI.

Invoke the pnic script using the command syntax **pnic** *command*; the OS environment variables are set such that this command can be executed from any path.

CLI Commands

CLI commands are given in Command Line Reference on page 79.

Editing Dynamic Rules with the CLI

Dynamic rules are stored in the file *rules.custom* in the */usr/local/pnic/0* directory.

To edit dynamic rules:

Step	Task
1	Change directories to */usr/local/pnic/0.*
2	Enter the command **vi rules.custom to edit dynamic rules** (see Appendix D, on page 125 for information on *vi*).
3	Enter rules according to the format described in “Writing Rules” on page 63.
4	Save your changes and exit *vi*.
5	Enter **pnic compilerules** to compile the new dynamic rules.
6	Enter **pnic loadrules** upload the dynamic rules to the FPGA.

MAC Rewriting

The MAC rewrite feature allows the least significant byte (LSB) of a packet’s destination MAC address to be overwritten with a user-specifed value. This feature may be used to load balance or redirect traffic.

This feature can be enabled per channel. When MAC rewrite is enabled, the P10 appliance classifies the incoming traffic into one of 256 hash buckets to determine the value to be written to the LSB of destination MAC address. A hash function based on the source and destination IP addresses is used to calculate an 8-bit index for each incoming packet. The index is used to look up the LSB values to be written into the packet.

To enable MAC rewriting:

Step	Task
1	Enter the command **pnic macrewrite-on 0** *channel* to enable MAC rewriting.
2	Verify that MAC rewrite is enabled using the command **pnic showconf**.

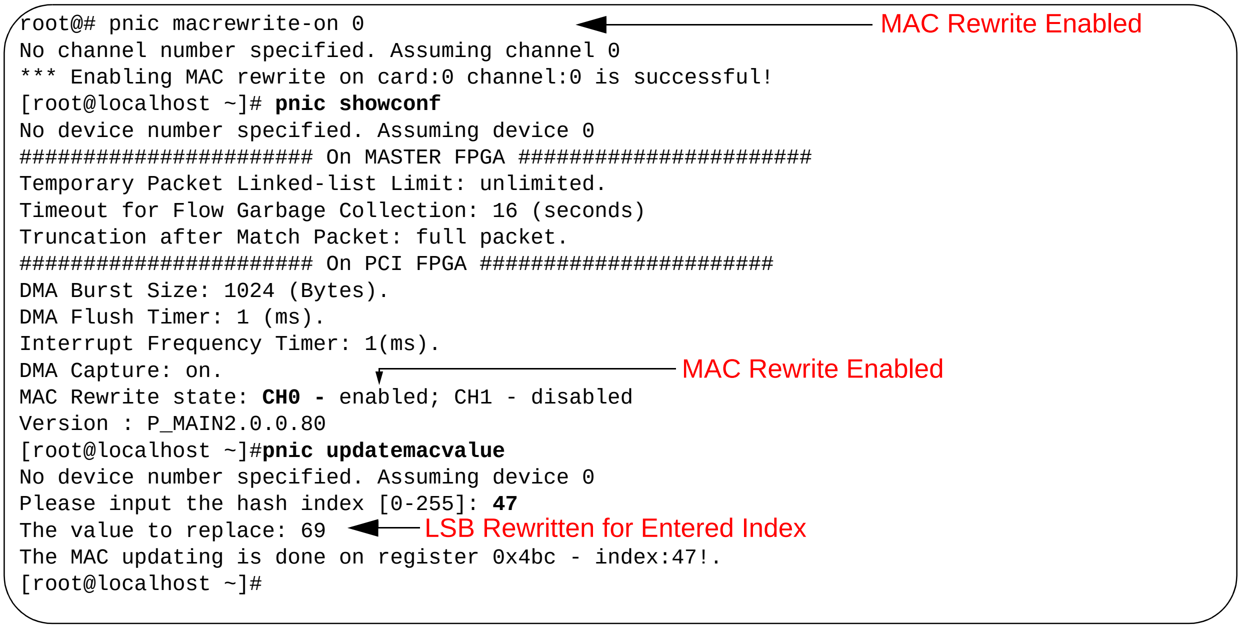
Two additional commands are available with this feature:

- **pnic updatemacvalue**—Assigns a new LSB for a particular index.
- **pnic getmachasindex**—Obtains the hash index value for a particular source and destination IP combination.

In Figure 34:

1. MAC rewriting is enabled
2. The user associates an LSB value with a particular index value.
3. All packets with source and destination IP addresses that hash to this index value then have the the least significant byte of their destination MAC address overwritten with the user-entered LSB value.

Figure 34 Rewriting Destination MAC Addresses to Load Balance



```
root@# pnic macrewrite-on 0
No channel number specified. Assuming channel 0
*** Enabling MAC rewrite on card:0 channel:0 is successful!
[root@localhost ~]# pnic showconf
No device number specified. Assuming device 0
####################### On MASTER FPGA #######################
Temporary Packet Linked-list Limit: unlimited.
Timeout for Flow Garbage Collection: 16 (seconds)
Truncation after Match Packet: full packet.
####################### On PCI FPGA #######################
DMA Burst Size: 1024 (Bytes).
DMA Flush Timer: 1 (ms).
Interrupt Frequency Timer: 1(ms).
DMA Capture: on.
MAC Rewrite state: CH0 - enabled; CH1 - disabled
Version : P_MAIN2.0.0.80
[root@localhost ~]#pnic updatemacvalue
No device number specified. Assuming device 0
Please input the hash index [0-255]: 47
The value to replace: 69
The MAC updating is done on register 0x4bc - index:47!.
[root@localhost ~]#
```

Removing VLAN Tags

The P-Series can strip the VLAN tag from incoming packets before they exit the egress port. Enable the feature using the command **pnic vlan-remove-enable**. The frame CRC is recalculated when this feature is enabled. If an incoming packet is untagged, it is not changed.

View the enable state of this feature using the command **pnic showconf**.

Chapter 8 Compiling Rules

The *P-Series Network Interface Card Compiler (pnic-Compiler)* produces user-defined firmware for the appliances. The user-defined input is a set of signature-based rules in Snort syntax, and compilation directives. The output of the compiler is a Xilinx bit file and ASCII mapping files that map specified signatures to internal configuration registers. The configuration registers are used to disable/enable rules or block packets.

Creating Rules Files

Store rules files in a *pnic-compiler* sub-directory — for example *pnic-compiler/rules*. Force10 recommends not storing rules files elsewhere because this increases the length of the firmware file name.

Rules Capacity

The maximum rules capacity for the P10 is approximatly 14000 static rules or 200 dynamic rules. The space required for a static rule depends upon its complexity.

Compiling Rules

➡ **Note:** The pnic-Compiler is managed with GNU make.

To complile rules:

Step	Task
1	Change directory to *pnic-compiler.*
2	Enter the command **gmake***.* This command invokes the configuration script, the pnic-Compiler, and the Xilinx compiler, in succession. Entering **time gmake** invokes the same processes, but this command measures the compilation time as well.
3	The script prompts you for a number of compilation options. Refer to Table 8 for a description of each option, and enter a response for each.

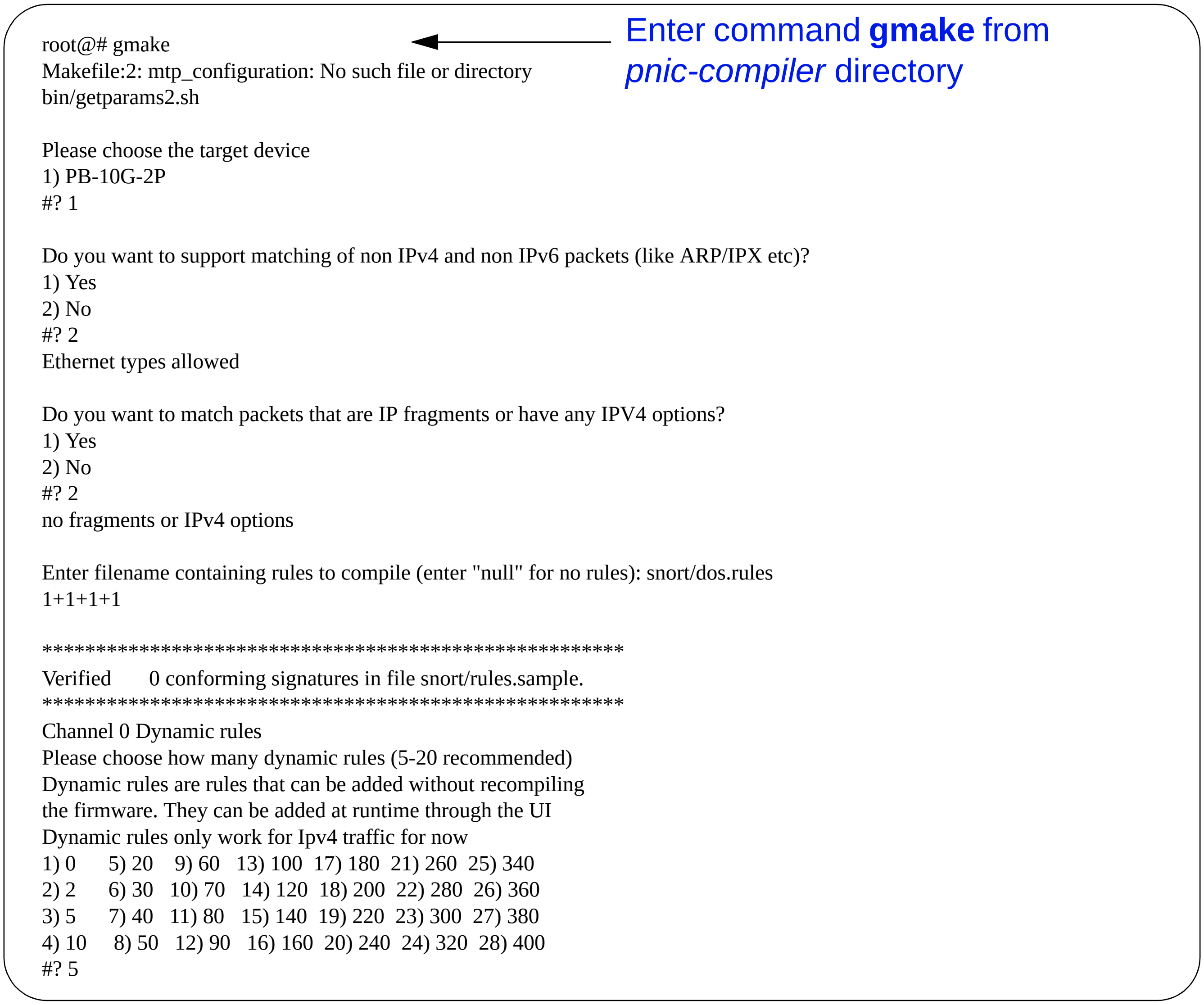
Table 8 Compiler Configuration Options

	Compilation Option	Description
1	Target Device	Choose the model of your appliance. • The P10 requires type **PB-10G-2P** (see Figure 35 on page 58)
2	Match non-IP Traffic	Answering **Yes** to this option matches packets that are not IPv4. This option should be set to **No** if only IP traffic is allowed. (see Figure 35 on page 58)
3	Match Fragmented IPv4 Packets or IPv4 Packets w/ Options	Answering **Yes** to this option: • Adds a rule to match fragmented IPv4 packets • Adds a rule to match IPv4 packets with any option in the header (see Figure 35 on page 58).
4	Rules File	Specify the rules file that contains the Snort rules that will be compiled into firmware. • Include the relative path of the file in your entry. • Your entry is used to create the firmware names. • Enter **null** to create firmware with no static rules; compiling firmware with no static rules maximizes dynamic rule capacity (see Figure 35 on page 58). **Note:** The script performs a syntax check on the input file. If there are errors, you are prompted to enter the file name again. The entry must be made at the prompt; if the *Enter* key is pressed erroneously such that the entry cannot made at the prompt, enter **Ctrl-C** to halt the configuration process, and then enter **gmake** to begin again.
5	Dynamic Rules	Enter the number of dynamic rules to synthesize. • If you enter one of the sample Snort rules files, choose the minimum number of dynamic rules; otherwise, the placing may fail. • If you are using fewer static rules, you can increase the number of dynamic rules up to approximately 30 for each channel (60 in total) (see Figure 35 on page 58). **Note:** The number of dynamic rules specified in this option is guideline that the compiler uses to reserve space on the FPGA. The number you choose is the approximate number of rules you will be able to configure at runtime. The amount of space a rule consumes varies based on the complexity of the rule. Therefore, you might not be able to compile as many dynamic rules as specified in this option if the rules are complex.
6	meta.rules	The pnic-Compiler prepends a set of fixed rules called *meta.rules* — located in the *pnic-compiler/rules* directory. The rules in this file report on flow information and provide compatibility with Snort; include or exclude this file considering that including them allows you to run Snort on the DPI interface. It is best to include this file if Snort is being used as the front end. If not using Snort as the front end, these rules should not be included or they should be changed to accommodate other packet analysis requirements (see Figure 36 on page 59).

Table 8 Compiler Configuration Options

	Compilation Option	Description
7	Segmentation Evasion Rules	The pnic-Compiler prepends a set of fixed rules—called *evasion.rules* — located in the *pnic-compiler/rules* directory. The rules help detect attacks which are using strategic TCP segmentation to avoid detection. It is best to include this file if Snort is being used as the front end. If not using Snort as the front end, these rules should not be included or they should be changed to accommodate other packet analysis requirements (see Figure 36 on page 59).
8	Maximum String	Specify the maximum number of bytes a single static rule can use for content matching. A low value truncates the match string and increases the number of rules that can fit into the FPGA, but this is at the expense of increased false positives. A value lower than 1024 is not recommended unless you can cope with the increased number of false positives through Snort or some other means (see Figure 37 on page 60).
9	Firmware Name	Enter a mnemonic name for the firmware you are about to create.
10	Confirmation	Enter **Yes** to save the configuration and compile the Snort rules into firmware (see Figure 37 on page 60).

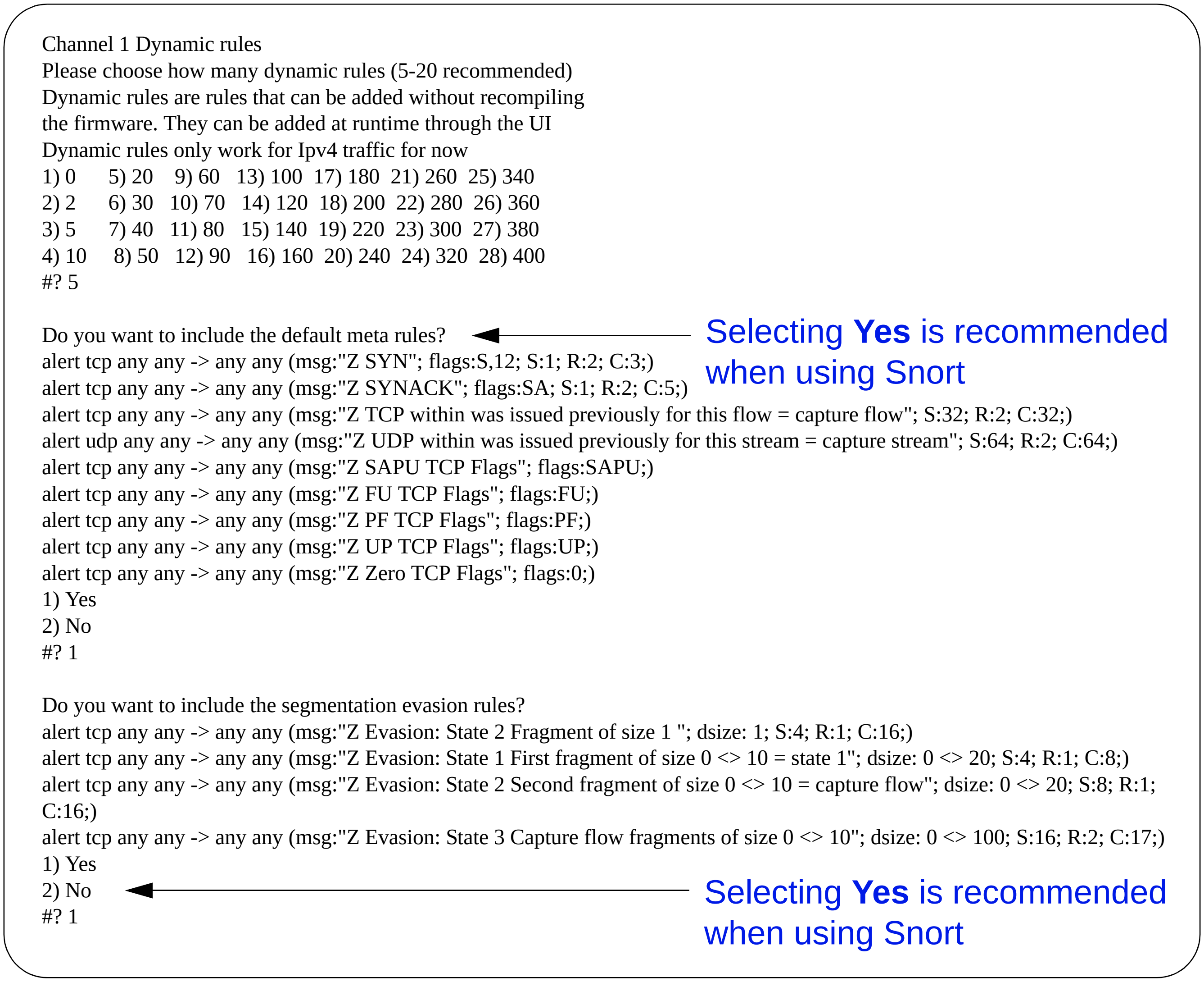
Figure 35 pnic-Compiler Option 1-6



```
root@# gmake
Makefile:2: mtp_configuration: No such file or directory
bin/getparams2.sh

Please choose the target device
1) PB-10G-2P
#? 1

Do you want to support matching of non IPv4 and non IPv6 packets (like ARP/IPX etc)?
1) Yes
2) No
#? 2
Ethernet types allowed

Do you want to match packets that are IP fragments or have any IPV4 options?
1) Yes
2) No
#? 2
no fragments or IPv4 options

Enter filename containing rules to compile (enter "null" for no rules): snort/dos.rules
1+1+1+1

*****************************************************
Verified      0 conforming signatures in file snort/rules.sample.
*****************************************************
Channel 0 Dynamic rules
Please choose how many dynamic rules (5-20 recommended)
Dynamic rules are rules that can be added without recompiling
the firmware. They can be added at runtime through the UI
Dynamic rules only work for Ipv4 traffic for now
1) 0     5) 20    9) 60   13) 100  17) 180  21) 260  25) 340
2) 2     6) 30   10) 70   14) 120  18) 200  22) 280  26) 360
3) 5     7) 40   11) 80   15) 140  19) 220  23) 300  27) 380
4) 10    8) 50   12) 90   16) 160  20) 240  24) 320  28) 400
#? 5
```

Enter command **gmake** from *pnic-compiler* directory

Figure 36 pnic-Compiler Option 6-7

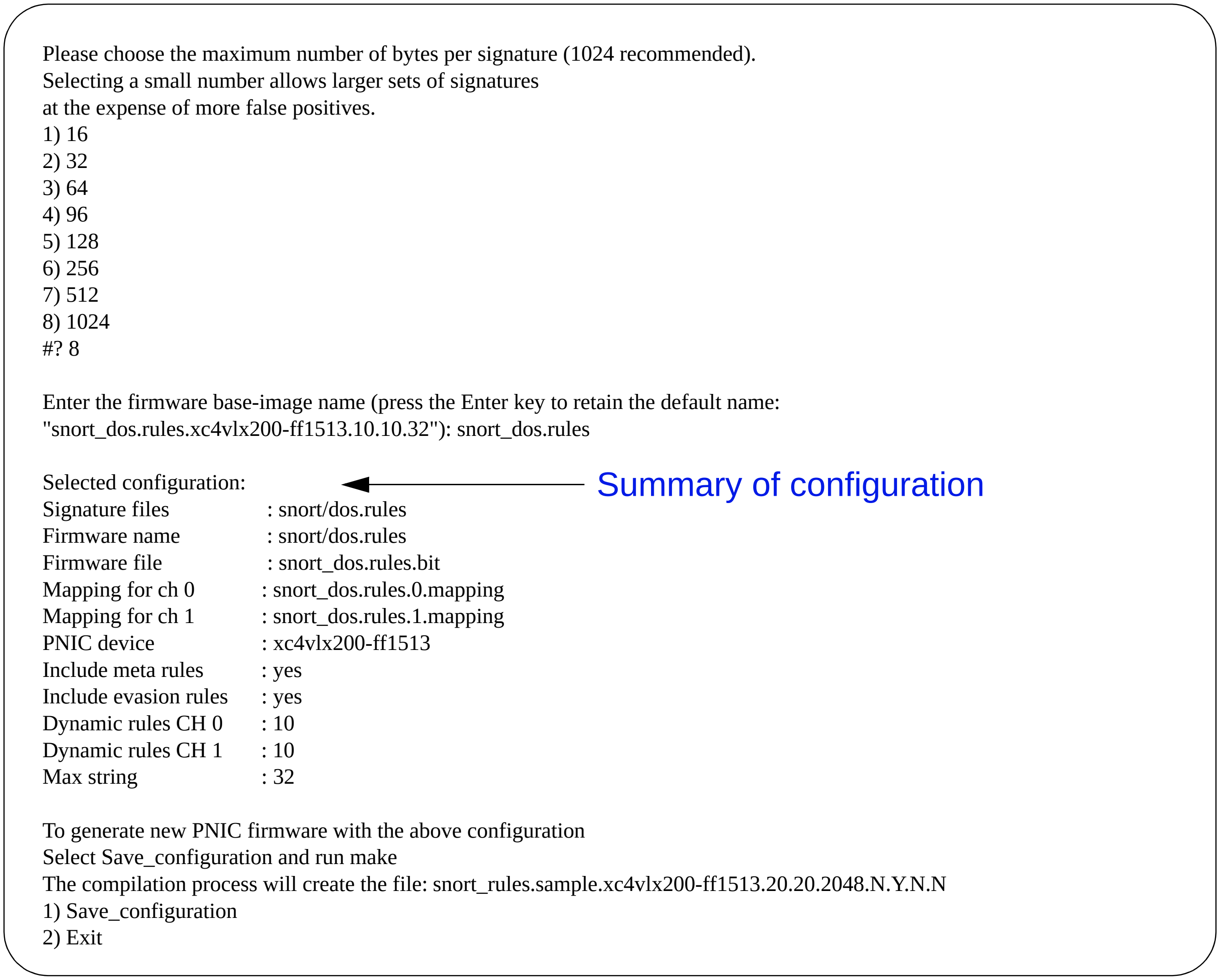


```
Channel 1 Dynamic rules
Please choose how many dynamic rules (5-20 recommended)
Dynamic rules are rules that can be added without recompiling
the firmware. They can be added at runtime through the UI
Dynamic rules only work for Ipv4 traffic for now
1) 0     5) 20    9) 60   13) 100  17) 180  21) 260  25) 340
2) 2     6) 30   10) 70   14) 120  18) 200  22) 280  26) 360
3) 5     7) 40   11) 80   15) 140  19) 220  23) 300  27) 380
4) 10    8) 50   12) 90   16) 160  20) 240  24) 320  28) 400
#? 5

Do you want to include the default meta rules?
alert tcp any any -> any any (msg:"Z SYN"; flags:S,12; S:1; R:2; C:3;)
alert tcp any any -> any any (msg:"Z SYNACK"; flags:SA; S:1; R:2; C:5;)
alert tcp any any -> any any (msg:"Z TCP within was issued previously for this flow = capture flow"; S:32; R:2; C:32;)
alert udp any any -> any any (msg:"Z UDP within was issued previously for this stream = capture stream"; S:64; R:2; C:64;)
alert tcp any any -> any any (msg:"Z SAPU TCP Flags"; flags:SAPU;)
alert tcp any any -> any any (msg:"Z FU TCP Flags"; flags:FU;)
alert tcp any any -> any any (msg:"Z PF TCP Flags"; flags:PF;)
alert tcp any any -> any any (msg:"Z UP TCP Flags"; flags:UP;)
alert tcp any any -> any any (msg:"Z Zero TCP Flags"; flags:0;)
1) Yes
2) No
#? 1

Do you want to include the segmentation evasion rules?
alert tcp any any -> any any (msg:"Z Evasion: State 2 Fragment of size 1 "; dsize: 1; S:4; R:1; C:16;)
alert tcp any any -> any any (msg:"Z Evasion: State 1 First fragment of size 0 <> 10 = state 1"; dsize: 0 <> 20; S:4; R:1; C:8;)
alert tcp any any -> any any (msg:"Z Evasion: State 2 Second fragment of size 0 <> 10 = capture flow"; dsize: 0 <> 20; S:8; R:1;
C:16;)
alert tcp any any -> any any (msg:"Z Evasion: State 3 Capture flow fragments of size 0 <> 10"; dsize: 0 <> 100; S:16; R:2; C:17;)
1) Yes
2) No
#? 1
```

Selecting **Yes** is recommended when using Snort

Selecting **Yes** is recommended when using Snort

Figure 37 pnic-Compiler Option 8-9



```
Please choose the maximum number of bytes per signature (1024 recommended).
Selecting a small number allows larger sets of signatures
at the expense of more false positives.
1) 16
2) 32
3) 64
4) 96
5) 128
6) 256
7) 512
8) 1024
#? 8

Enter the firmware base-image name (press the Enter key to retain the default name:
"snort_dos.rules.xc4vlx200-ff1513.10.10.32"): snort_dos.rules

Selected configuration:
Signature files          : snort/dos.rules
Firmware name            : snort/dos.rules
Firmware file            : snort_dos.rules.bit
Mapping for ch 0        : snort_dos.rules.0.mapping
Mapping for ch 1        : snort_dos.rules.1.mapping
PNIC device             : xc4vlx200-ff1513
Include meta rules      : yes
Include evasion rules   : yes
Dynamic rules CH 0      : 10
Dynamic rules CH 1      : 10
Max string              : 32

To generate new PNIC firmware with the above configuration
Select Save_configuration and run make
The compilation process will create the file: snort_rules.sample.xc4vlx200-ff1513.20.20.2048.N.Y.N.N
1) Save_configuration
2) Exit
```

Summary of configuration

Starting and Stopping the pnic-Compiler

Enter the keyboard command **Ctrl-C** or a *SIGINT* signal to interrupt the compilation or configuration process. Enter **gmake** to restart the process from where it was interrupted. The compilation process restarts at the point where it was halted; the configuration process restarts from the beginning.

During compilation, enter **Ctrl-C** followed by **gmake clean** to regenerate firmware with different options. This erases the current configuration and resets the compilation process. Previously generated firmware files are not erased.

Configuration and Generated Files

Table 9 describes the files that are used or generated by the pnic-Compiler.

Table 9 Configuration and Generated Files

File	Description	Location
pnic_*.bit	Generated after compiling static rules. They are then renamed and copied to /usr/local/pnic/firmware. When selecting firmware, the *.bit* files are symbolically linked to the corresponding renamed files in the firmware directory.	/usr/local/pnic/0
pnic_*.mapping	Generated after compiling static rules. They are then renamed and copied to /usr/local/pnic/firmware. When selecting firmware, the *.mapping* files are symbolically linked to the corresponding renamed files in the firmware directory.	/usr/local/pnic/0
<firmware_filename>.bit	Firmware files for Channel 0 and Channel 1. They are the renamed *.bit* files that were generated after compiling static rules. When selecting firmware, these are the files to which the *.bit* files in */usr/local/pnic/0* are symbolically linked.	/usr/local/pnic/firmware/<firmware_filename>
<firmware_filename>.mapping	Firmware files for Channel 0 and Channel 1. They are the renamed *.mapping* files that were generated after compiling static rules. When selecting firmware, these are the files to which the *.mapping* files in */usr/local/pnic/0* are symbolically linked.	/usr/local/pic/firmware/<firmware_filename>
pnic_*.bin	Contain compiled dynamic rules for Channel 0 and Channel 1.	/usr/local/pnic/0
pnic_*.custmapping	Contain the capture/forward policies for each rule on Channel 0 and Channel 1.	/usr/local/pnic/0
rules.custom	Contains dynamic rules written in Snort syntax.	/usr/local/pnic/0

Firmware Filenames

The pnic-Compiler creates new firmware — in the */usr/local/pnic/firmware* directory — consisting of four *.bit* files and eight *.mapping* files.

The default firmware filenames follow a naming convention designed to identify three properties:

- The appliance that can use it
- The number of dynamic rules
- The maximum allowed number of half-bytes per rule

Firmware files have the format:

```
<name>.<type>.<dynamic{0|1}>.<maxstring>.{0|1}.{bit|mapping}
```

Table 10 describes each of the elements in this format.

Table 10 Firmware Filename Description

Element	Description
<name>	This field is a mnemonic name identifying the original rules file you supplied during the compilation of the firmware.
<type>	This field identifies the card type. The P10 is represented by *xc4vlx200-ff1513.*
<dynamic{0|1}>	This field is the estimated number of dynamic rules that you can enter at runtime for the two channels.
<maxstring>	This field is the maximum number of half-bytes the compiler allocates for each rule. A typical value is 2048 to indicate that the compiler truncates match string to 1024 bytes. Typically a value is 2048, which does not result in any truncation. Lower values are possible and result in a larger number of rules, but this increases the probability of false positives for rules with truncated match strings.
{0|1}	This field indicates whether the file is for Channel 0 or Channel 1.
{bit|mapping}	The compiling process generates 12 files which together make firmware. 8 files have the extension *.mapping*, and 4 have the extension *.bit.*

Compiler Errors

- If too many dynamic rules are specified in Option 9 of the compiler configuration phase, the compilation process fails, and you receive a “Error-PhysDesignRules” error message. In this case, enter **gmake clean** to erase the current configuration and begin again.
- If too many rules stored in the rules file specified in Option 6 of the compiler configuration phase, the compilation process fails. In this case, enter **gmake clean** to erase the current configuration and begin again.

Chapter 9 Writing Rules

P-Series rule syntax is based on Snort. Both rule structures are described in this chapter.

- Snort Rule Syntax on page 63
- P-Series Rule Syntax on page 66

Snort Rule Syntax

Snort rules are descriptions of traffic plus a prescribed action that is taken if a packet matches that description. Rules are divided into two sections:

- *Header*: The header contains the action, protocol, source and destination IP addresses (with subnet masks), and the source and destination ports.
- *Options:* The options section contains alert messages, and specifies values to search for inside the packet.

Table 11 shows the syntax for Snort rules, and Table 12 shows an example. The text preceding parenthesis is the header, and the section enclosed in parenthesis contains the rule options. The words before the colons in the rule options section are option keywords. Rules that span multiple lines must have a backslash at the end of the line. All rules and options must be punctuated with a semicolon.

Table 11 Snort Rule Syntax

action protocol source_address source_port -> *destination_address destination_port*\ (content:”*data_string*”; msg:”*message*”);

Table 12 Snort Rule Example

<table>
<tr><td>alert tcp any any -> 192.168.1.0/24 111 (content:”| 00 01 86 a5 |”; msg:”mounted access”);</td></tr>
</table>

Snort Rule Headers

Action

The first item in a rule is the action keyword. It dictates how Snort is to handle a packet that matches the rule. All of the elements in a rule must be true for Snort to execute the action. There are five actions keywords in Snort:

- **alert** directs Snort to generate an alert and log the packet.
- **log** directs Snort to log the packet.

- **pass** directs Snort to ignore the packet.
- **activate** directs Snort to generate an alert and activate another specified rule.
- **dynamic** directs Snort to disregard the rule until it is activated by another rule. Once activated, the action defaults to log.

Note: The default actions for the P-Series are different from Snort. See “P-Series Rule Syntax” on page 66. The meaning of the Snort action keyword dynamic is not the same as P-Series dynamic rules. Dynamic rules in Snort are rules that must be activated, where as with the P-Series, dynamic rules are any rules that are uploaded to the FPGA without creating new firmware.

Protocol

Snort supports four protocols: **tcp**, **udp**, **icmp**, or **ip**. The protocol keyword follows the action keyword.

Source Addresses

The source address and port follow the protocol keyword. Addresses are written using dotted-decimal notation with the subnet mask in CIDR block notation. For example, the address/CIDR combination 192.168.1.0/24 signifies a block of addresses from 192.168.1.1 to 192.168.1.255. The keyword *any* may be used to define any source address.

The address field can be negated by placing an exclamation point before the address. This operator specifes all addresses other than the one contained in the rule. The rule in Table 13 indicates specifes all traffic originating from outside the local network and destined for the local network.

Note: The negation operator may not be placed before the keyword any.

Table 13 Rules Containing Address Negation

```
alert tcp !192.168.1.0/24 any -> 192.186.1.0/24 111(content:”| 00 01 86 a5 |”; msg:”mounted access”;)
```

Lists of IP addresses can be specified by placing the addresses in brackets and separating each address with a comma; do not include spaces. Table 14 shows an example of a rule containing multiple addresses.

Table 14 Rules Containing Multiple IP Addresses

```
alert tcp ![192.168.1.0/24,10.1.1.0/24] any -> [192.186.1.0/24,10.1.1.0/24] 111(content:”| 00 01 86 a5 |”;\
msg:”mounted access”;)
```

Ports

Port numbers may be specified by the keyword *any*, a single port number, ranges, and by negation. *any* specifies any port. Static ports are indicated by a single port number, for example, 23 for Telnet. Port ranges can be specified using a colon as a range operator. It can be applied in three ways, as shown by Table 15.

Table 15 Rules Containing the Port Number Range Operator

```
log udp any any -> 192.168.1.0/24 1:1024 log udp
log tcp any any -> 192.168.1.0/24 :6000
log tcp any :1024 -> 192.168.1.0/24 500:
```

- A colon between two port numbers indicates all ports between those ports, including the specified ports.
- A colon before a port number indicates all ports less than or equal to the specified port.
- A colon after a port number indicates all ports greater than or equal to the specifed port.

The negation operator can also be used in combination with port numbers. The rule in Table 16 logs all TCP traffic destined for ports other than port 6000 on the local network.

Table 16 Rules Containing the Port Number Negation Operator

```
log tcp any any -> 192.168.1.0/24 !6000:6000
```

Note: The negation operator may not be placed before the keyword any. The ICMP protocol does not require a port number.

Direction Operator

The direction operator, **->**, indicates direction of the traffic to which the rule applies. The source IP address and port are on the left side of the direction operator, and the destination address and port are on the right side of the operator.

There is also a bidirectional operator, **<>**. This directs Snort to consider traffic originating from either of the specified addresses and ports. This operator can be used for analyzing both sides of a conversation. An example of the bidirectional operator being used to record both sides of a Telnet session is shown in Table 17.

Table 17 Rules Containing the Bidirectional Operator

```
log tcp !192.168.1.0/24 any <> 192.168.1.0/24 23
```

Destination Address and Port

The destination address and port follow the direction operator. The syntax of these parameters are the same as the source address and port. See "Source Addresses" on page 64, and "Ports" on page 65.

Snort Rule Options

Options are made of a keyword and an argument. An argument is the packet data against which the rule is matched. Option keywords are followed by a colon, and each option is puncutated with a semi-colon. Table 19 lists the option keywords that the P-Series supports.

P-Series Rule Syntax

P-Series rules have a syntax that is slightly different from Snort rules. P-Series rules have the following syntax:

capture/forward_policy **on** *channel Snort_rule*

- *capture/forward* policy can have four values: *alert*, *permit*, *divert*, or *deny*. These settings are described in Table 5 on page 28.
- *channel* can be **c0** for Channel 0, **c1** for Channel 1, or **all** for both channels.
- *Snort_rule* is a rule written in Snort syntax.

Table 18 shows an example P-Series rule.

Table 18 P-Series Rule Example

```
alert on c1 any any -> any any (msg:"Z Default rule fragmented ip";)
```

Note: P-Series does not support the Snort action keywords *log*, *pass*, *activate*, and *dynamic*. P-Series supports the action keywords *alert*, *permit*, *divert*, and *deny*.

P-Series Supported Snort Keywords

Table 19 lists Snort keywords that the P-Series supports for both dynamic and static rules.

Table 19 Supported Snort Keywords for Static and Dynamic Rules

Keyword	Static	Dynamic
ack	Yes	Yes
content	Yes, no negative.	No

Table 19 Supported Snort Keywords for Static and Dynamic Rules

Keyword	Static	Dynamic
depth	No	No
dsize	Yes	No
flags	Yes	Yes, no wild card
flow	Yes	No
fragbits	Yes	No
fragoffset	Yes	No
icmp_id	Yes	Yes
icmp_seq	Yes	Yes
icode	Yes	Yes
id	Yes	Yes
ip_proto	Yes	Yes
itype	Yes	Yes
offset	No	No
nocase	Yes	No
protocol	ICMP, UDP, TCP, IP	ARP, ICMP, UDP, TCP, IP
seq	Yes	Yes
source address	Yes	Only /8/16/24/32 masks
destination address	Yes	Only /8/16/24/32 masks
source port	Yes	Yes, no ranges
destination port	Yes	Yes, no ranges
tos	Yes	Yes
ttl	Yes	Yes
uricontent	Yes, no negative.	No
window	Yes	No
within	No	No

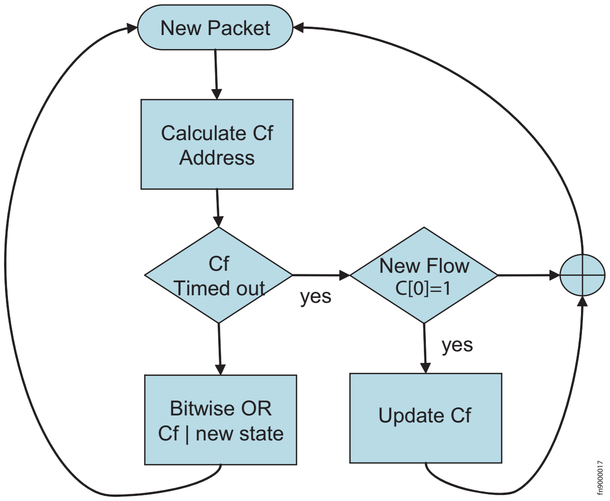
Writing Stateful Rules

Stateful matching improves the accuracy of detection because it adds ordering when specifying behaviors across multiple matching events. State transitions in the P-Series follow a non-cyclic pattern; no state transitions may erase any of the previous states. New state transitions are simply recorded via a non-destructive, additive operation.

As new states are produced, they are bitwise “*OR*-ed” with the current states contained in the per-flow register C_f., which is 16 bits wide. This method is different from stateful matching in software systems, where old state is removed after a set amount of time. It allows a deterministic wire-speed state management algorithm while guaranteeing that no match events are ever lost due to resource constraints.

Figure 38 shows the state matching algorithm. Note that the only time some state is erased is in the case of a timeout.

Figure 38 State Management Algorithm



Stateful Matching

Each signature *i* contains a pattern matching expression m_i that is compared to the incoming data stream in real time (time *t*). In addition, each signature may contain - at your discretion - three values, *s*, *c*, and *r*, which respectively specify:

- The pre-match state condition necessary for the signature to match (in addition to m_i)
- The post-match state condition applied after the signature has matched
- A directive indicating what to do with the matched packet

The *s* and *c* values are used to manage a per-flow register C_f, where the subscript *f* is the flow, or *sub-stream*, and the *r* value is used to direct the packet storage.

Pre-match Condition — the S Value

The value in register C_f is presented to all the signatures simultaneously during matching.

C_f must have all the bits specified by s_i (in addition to matching m_i) in order for the signature *i* to trigger. In other words, if the result of the logical “AND” of register C_f with s_i is non-zero and equal to s_i, the signature is allowed to trigger. Otherwise the signature is not triggered. Therefore value s_i is referred to as the pre-match bit pattern.

Post-match Condition — the C Value

The c_i value is the post-match bit pattern defined by the signature *i*. If m_i matches in the data stream, and the pre-match condition is met, c_i is logically “*OR*-ed” with the existing value in register C_f, and the result is written back to C_f.

In general for each signature *i* at time *t:*

$$If\ \left\{m_i \wedge (s_i^t\ \&\ C_f^{t-1}) = s_i^t\right\},\ then\ cp_i^t = c_i,\ else\left\{cp_i^t = 0\right\} \qquad Equation\ 3$$

$$C_f^t = \sum cp_i^t \Big| C_f^{t-1} \qquad Equation\ 4$$

where /\ is a logical “AND” operator, *&* is a bitwise AND, Sigma is a bit-wise “OR” of several terms, and | is a bitwise OR of two terms.

Equation 3 states that if there is a match $m_{i,}$ and the pre-match condition holds, the post-match condition cp_i is enabled.

Equation 4 states that at each cycle, the register C_f is updated by the bitwise OR of all the cp_i values of all the signatures, and a final bitwise OR with the previous state.

When a stateful flow is older than a timeout value, $C_f^{(t-1)}$ is ignored. It is replaced by 0x1. So, the rule for the first state of a flow should have s=1.

Packet Handling — the R Value

The constant r_i is a flag that tells the hardware what to do with a packet that has been matched to signature *i*. The memory used to store the matched packets is divided into *Temporary Memory* and *Match Memory.* If a packet is stored in Match Memory, action is requested from the host to process the matched packet. If a packet is stored in Temporary Memory, no action is requested from the host, as this represents only a partial match.

When a packet is stored in either Temporary Memory or Match Memory, a pointer to the previously stored packet in the same flow (contained in a portion of the flow register C_f) is also stored. Thus a packet stored in Match Memory may reference another packet stored in Temporary Memory, which in turn may reference more packets, thus forming a linked list of partial matches, starting with a packet stored in Match Memory.

The values for r_i have the following meanings:

1: store the packet in Temporary Memory

2: store the packet in Match Memory and notify host software

Note: If the Hash key option is selected, the R=2 flag no longer causes the packet to be stored in Temporary Memory.

Stateful Rule Examples

Table 20 Stateful Matching Signatures

Signature 1: alert on c0 tcp any any -> any any (msg:"SYN"; flags:S; S:1; R:0; C:3;)
Signature 2: alert on c0 tcp any any -> any any (msg:"ack"; flags:A+; S:2; R:1; C:4;)
Signature 3: alert on c0 tcp any any -> any any (msg:"ack"; flags:A+; S:4; R:2; C:4;)
Signature 4: alert on c0 tcp any any -> any any (msg:"frag"; dsize: 0 <> 100; S:1; R:1; C:9;)
Signature 5: alert on c0 tcp any any -> any any (msg:"frag"; dsize: 0 <> 100; S:8; R:1; C:16;)
Signature 6: alert on c0 tcp any any -> any any (msg:"frag"; dsize: 0 <> 100; S:16; R:2; C:16;)

In Table 20:

- Signature 1 matches any TCP SYN packet, erasing any expired C_f register; if this signatures triggers - meaning a SYN is present — it sets bits 0 and 1 (value 3) in the C_f register. The SYN packets is discarded (R=0).
- Signature 2 triggers if Signature 1 has triggered (the C_f register having bit 1 set) and a TCP packet contains an ACK bit. The result for this match is that bit 2 (value 4) is set in the $\mathrm{C_f}$ register. The packet is stored in Temporary Memory (R=1).
- Signature 3 triggers if Signature 2 has triggered (the C_f register having bit 2 (value 4) set) and another later TCP packet contains an ACK bit. The result for this match does not modify the existing content of the C_f register. The packet is stored in Match Memory, referencing the packet of Signature 2. The DPI driver then presents to the host the packet matched by 2, followed by the packet matched by 3, through the DPI network interface.

You can inspect Signatures 4, 5, and 6, and verify that they trigger a match and place a packet in Match Memory — thus alerting the host — if three consecutive packets are seen with size between 0 and 100. The third packet references the previous two stored in Temporary Memory. Thus, once the third packet is received, the three segments are presented to the host through the DPI network interface. Notice that the bit pattern used in the two rules avoids collision with the previous rule if the flow hashing also happens to collide.

The *meta.rules* File

The *meta.rules* file — located in the *pnic-compiler/rules* directory — specifies a number of stateful rules to be used with standard Snort rules (which use the *Flow* keyword). In addition, these rules implement a stateful mechanism to circumvent some common forms of TCP IDS evasion. The meta rules are given in Appendix C, on page 123.

Support for Snort's *flow* Keyword

The two stateful rules in Table 21 initiate a new flow if a *SYN* or a *SYN-ACK* are seen. A Snort *flow-established* keyword is translated to S:4 and S:2 for client-to-server and server-to-client flows, respectively. These keywords are automatically inserted by the PNIC-Compiler when a flow-established keyword is encountered during compilation. You can also insert the keywords directly into your rules.

Table 21 Flow Established Rules

alert tcp any any -> any any (msg:"Z SYN"; flags:S,12; S:1; R:2; C:3;)
alert tcp any any -> any any (msg:"Z SYNACK"; flags:SA; S:1; R:2; C:5;)

Handling Segmentation Evasion

Tools like *fragroute* or *Nessus* are used to fragment the packet payload in several TCP segments in order to evade packet-based signature systems. The stateful rules in Table 22 detect the arrival of packets exhibiting an anomalous use of TCP segmentation.

The start of the state machine is prompted by a *SYN*; state 1 is reached if a packet of length greater than 0 but less than 20 is detected; state 2 is reached if a packet of length 1 is received right after a SYN or a second packet of length greater than 0 but less than 20 is detected; the final state is reached if a packet of a length between 0 and 100 is seen. This state diagram was derived from observing common fragmentation evasion patterns; it seems to catch most of them. More complex state diagrams can also be devised at your discretion.

Table 22 TCP Packets with Anomalous Segmentation

alert on c0 tcp any any -> any any (msg:"Z Evasion: State 2 Fragment of size 1 "; dsize: 1; S:4; R:1; C:16;)
alert on c0 tcp any any -> any any (msg:"Z Evasion: State 1 First fragment of size 0 <> 20 = state 1"; dsize: 0 <> 20; S:4; R:1; C:8;)
alert on c0 tcp any any -> any any (msg:"Z Evasion: State 2 Second fragment of size 0 <> 20 = capture flow"; dsize: 0 <> 20; S:8; R:1; C:16;)
alert on c0 tcp any any -> any any (msg:"Z Evasion: State 3 Capture flow fragments of size 0 <> 100"; dsize: 0 <> 100; S:16; R:2; C:16;)

Support for Snort's *within* Keyword

Many buffer-overflow detection rules use a *within* keyword that verifies that an end-of-line character is received within a certain number of bytes from the start of the session.

If the *within* statement is for a large number of bytes, the check needs to be performed across TCP segments. In this case, several packets must be captured to find the end-of-line character (or whatever the character might be). For this reason, *within* statements capture the entire flow.

The *within* statements are translated by the PNIC-Compiler upon setting the S:32 and S:64 bits. This causes two rules to trigger the capturing of TCP and UDP flows.

Table 23 shows two rules which trigger the capturing of TCP and UDP flows.

Table 23 Capturing TCP and UDP Flows

alert on c0 tcp any any -> any any (msg:"Z TCP within was issued previously for this flow = capture flow"; S:32; R:2; C:32;)
alert on c0 udp any any -> any any (msg:"Z UDP within was issued previously for this stream = capture stream"; S:64; R:2; C:64;)

Anomalous TCP Flags

Some TCP packets with anomalous flags are captured by default to provide scan detection software diagnosis information. Table 24 shows rules which were derived from the Snort scan pre-processor.

Table 24 TCP Packets with Anomalous Flags

alert on c0 tcp any any -> any any (msg:"Z SAPU TCP Flags"; flags:SAPU;)
alert on c0 tcp any any -> any any (msg:"Z FU TCP Flags"; flags:FU;)
alert on c0 tcp any any -> any any (msg:"Z PF TCP Flags"; flags:PF;)
alert on c0 tcp any any -> any any (msg:"Z UP TCP Flags"; flags:UP;)
alert on c0 tcp any any -> any any (msg:"Z Zero TCP Flags"; flags:0;)

The compiler also automatically produces rules that match all packets that are IP fragments or have IP options. These rules are not specified in the *pnic.meta* file as they can be more efficiently implemented by the compiler directly.

Chapter 10 Firewall

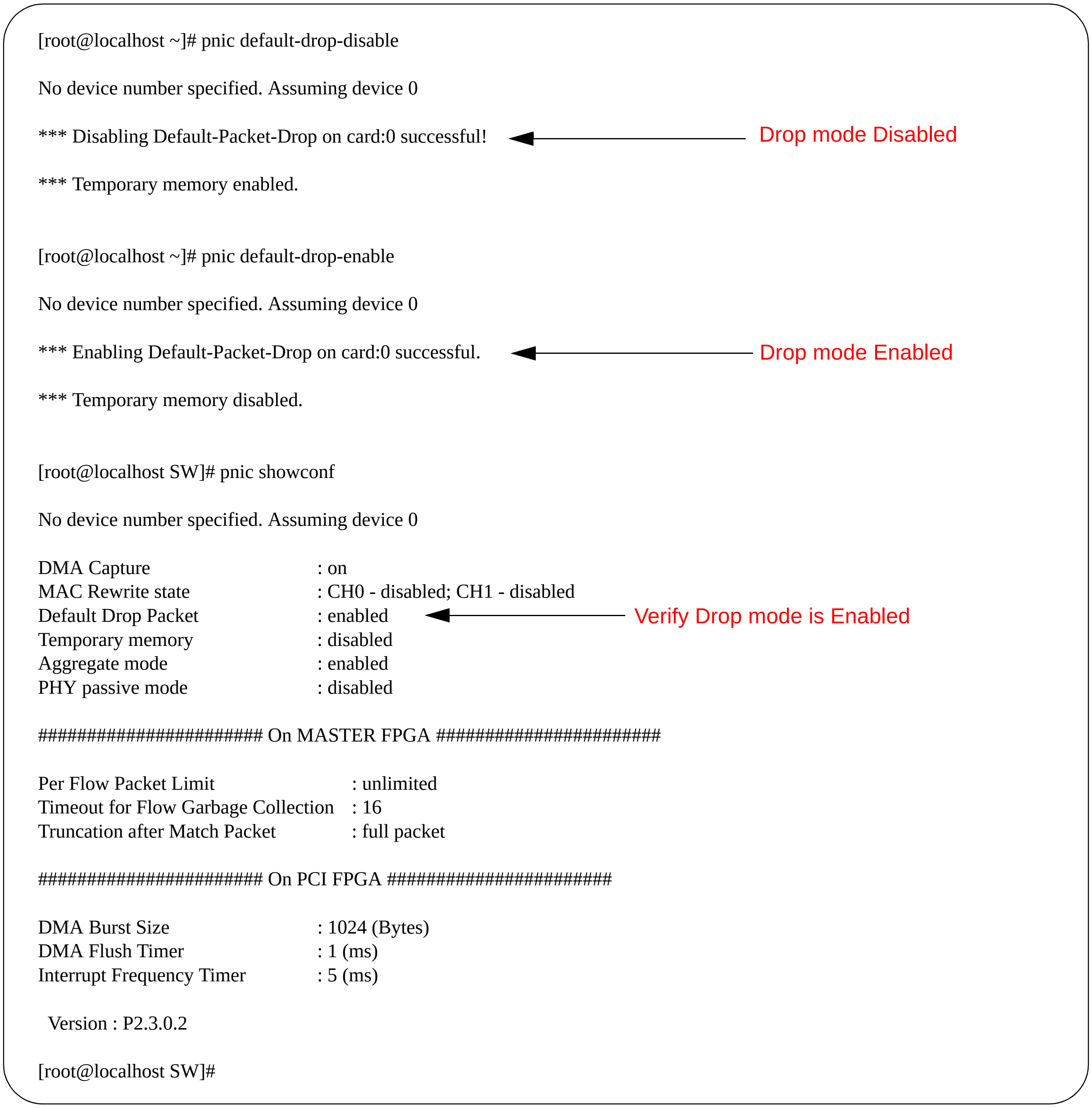
Deploying the P-Series as a Firewall

By default the P-Series is an IDS/IPS system; the P-Series forwards all traffic by default and blocks packets only if it matches a rule. You can deploy the P-Series as a limited firewall by enabling Drop mode. In Drop mode, the P-Series blocks all traffic by default and forwards traffic only if it matches a rule.

Enabling the Firewall

Enable Drop mode using the command **pnic default-drop-enable**. Disable Drop mode using the command **pnic default-drop-disable**. These commands are shown in Figure 39.

Figure 39 Enabling and Disabling Drop Mode



```
[root@localhost ~]# pnic default-drop-disable

No device number specified. Assuming device 0

*** Disabling Default-Packet-Drop on card:0 successful!

*** Temporary memory enabled.


[root@localhost ~]# pnic default-drop-enable

No device number specified. Assuming device 0

*** Enabling Default-Packet-Drop on card:0 successful.

*** Temporary memory disabled.


[root@localhost SW]# pnic showconf

No device number specified. Assuming device 0

DMA Capture                     : on
MAC Rewrite state               : CH0 - disabled; CH1 - disabled
Default Drop Packet             : enabled
Temporary memory                : disabled
Aggregate mode                  : enabled
PHY passive mode                : disabled

###################### On MASTER FPGA ######################

Per Flow Packet Limit               : unlimited
Timeout for Flow Garbage Collection : 16
Truncation after Match Packet       : full packet

###################### On PCI FPGA ######################

DMA Burst Size                  : 1024 (Bytes)
DMA Flush Timer                 : 1 (ms)
Interrupt Frequency Timer       : 5 (ms)

  Version : P2.3.0.2

[root@localhost SW]#
```

Allowing Traffic through the Firewall

To allow packets through the firewall you must write rules so that packets that you want the appliance to forward match those rules. Rules can be as simple as allowing traffic destined to a port. Stateful rules can be used to allow all traffic for an established connection. To allow non-IP traffic to pass through the firewall, you must select “Yes” for compiler option 2, as described in Table 8 on page 56.

Sample rules for a firewall deployment are available in file *pnic-compiler/rules/fw.rules*.

Writing Rules for a Firewall Deployment

Rules for a firewall deployment are written in the same Snort-based syntax as IDS/IPS rules. The difference is that you must describe packets that you want to forward, rather than block. See P-Series Rule Syntax on page 66.

In Table 25 stateful rules are used to allow specified traffic into the internal network. Notice that in the incoming direction, the policies require that the packet be destined to a set of allowed ports, while in the outgoing direction, there is no port requirement. This asymmetry produces typical firewall behavior.

The Drop mode can also accommodate arbitrary rules that do not assume an inside and outside interface. This is an attractive quality since the notion of inside and outside is often blurred in modern network topologies. Also note that traditional IPS and IDS rules can be coupled with the firewall rules to block packets and/or capture suspicious packets.

Table 25 Sample Firewall Rules

```
#permit: let through and do not log to the host
#alert: let through and log to the host
#deny: DO NOT let through and do not log to the host
#divert: DO NOT let through and log to the host

# S:<precondition>; C:<postcondition> R:<logging>
# A packet is matched if precondition matches the current state of that flow;
# in that case the postcondition is ORed and applied to rewrite the state of that flow;
# A precondition of 1 starts a new flow
# logging should be set to 2 for most cases; see the user manual for R:1

# Topology assumption
#c0 : Unsecured EXTERNAL network
#c1 : Secured INTERNAL network

# specify here your inside networks
#var INTERNAL 192.168.50.0/24
var INTERNAL any
var EXTERNAL any

# specify here your outside DNS servers
#var DNS [10.11.0.1, 10.11.0.2]
var DNS any
var DNSPORT 53

#specify here the services provided from the inside network
#var ALLOWEDPORTS [21,22,25]
var ALLOWEDPORTS 22

#allow INTERNAL network to poke a hole through the firewall for TCP services
permit on c1 tcp $INTERNAL any -> $EXTERNAL any (msg:"Z SYN"; flags:S; S:1; R:2; C:3;)
permit on c0 tcp $EXTERNAL any -> $INTERNAL any (msg:"Z SYNACK"; flags:SA; S:2; R:2; C:4;)

#allow EXTERNAL network to poke a hole through the firewall if accessing any of the ALLOWEDPORTS
permit on c0 tcp $EXTERNAL any -> $INTERNAL $ALLOWEDPORTS (msg:"Z SYN"; flags:S; S:1; R:2; C:3;)
permit on c1 tcp $INTERNAL $ALLOWEDPORTS -> $EXTERNAL any (msg:"Z SYNACK"; flags:SA; S:2; R:2; C:4;)

#allow TCP packets on the established flow/hole (INTERNAL <--> EXTERNAL)
permit on all tcp any any -> any any (msg:"Z TCP flow allowed"; S:4; R:2; C:4;)

#allow INTERNAL network to poke a hole through the firewall for DNS queries
permit on c1 udp $INTERNAL any -> $DNS $DNSPORT (msg:"DNS query"; S:1; R:2; C:9;)
permit on c0 udp $DNS $DNSPORT -> $INTERNAL any (msg:"DNS reply"; S:8; R:2; C:16;)

#allow UDP packets for the established UDP flow/holes (INTERNAL <--> DNS)
permit on all udp any any -> any any (msg:"Z UDP flow allowed"; S:16; R:2; C:16;)

#bad stuff; do not let though and do not log
deny on all tcp any any -> any any (msg:"Z SAPU TCP Flags"; flags:SAPU;)
deny on all tcp any any -> any any (msg:"Z FU TCP Flags"; flags:FU;)
deny on all tcp any any -> any any (msg:"Z PF TCP Flags"; flags:PF;)
deny on all tcp any any -> any any (msg:"Z UP TCP Flags"; flags:UP;)
deny on all tcp any any -> any any (msg:"Z Zero TCP Flags"; flags:0;)
```

Appendix A Command Line Reference

The command line interface (CLI) is an alternative to the GUI for managing the appliance. A script called *pnic* is used to perform the same management functions as the GUI.

Invoke the pnic script using the commands in this chapter; the OS environment variables are set such that these command can be executed from any path.

- pnic showconf on page 108
- pnic show-firmwares on page 108
- pnic showtech on page 109
- pnic start on page 110
- pnic stop on page 111
- pnic temp-mem-disable on page 112
- pnic temp-mem-enable on page 112
- pnic updatemacvalue on page 113
- pnic vlan-remove-disable on page 114
- pnic vlan-remove-enable on page 114
- pnic web-gui-start on page 115
- pnic web-gui-stop on page 116

Note: The P10 does not support multiple network interface cards. Therefore, the only valid entry for the *number* variable is 0.

Card 0 and Channel 0 are assumed for all commands if the *card* and *number* options are not specified.

pnic aggregate-mode-disable

Receive client-to-server and server-to-client traffic on different ports.

Syntax

pnic aggregate-mode-disable [*number*]

Enable aggregate mode using the command **pnic aggregate-mode-enable**.

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 40 pnic aggregate-mode-disable Command Example

```
[root@localhost SW]# pnic aggregate-mode-disable
No card number specified. Assuming card 0

*** Aggregate mode disabling on card:0 successful.

[root@localhost SW]#
```

Related Commands

pnic aggregate-mode-enable	Receive both client-to-server and server-to-client traffic on one port.

pnic aggregate-mode-enable

Receive both client-to-server and server-to-client traffic on one port. This is the default behavior.

Syntax

pnic aggregate-mode-enable [*number*]

Disable aggregate mode using the command **pnic aggregate-mode-disable**.

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 41 pnic aggregate-mode-enable Command Example

```
[root@localhost SW]# pnic aggregate-mode-enable
No card number specified. Assuming card 0

*** Aggregate mode enabling on card:0 successful.

[root@localhost SW]#
```

Related Commands

pnic aggregate-mode-disable	Receive client-to-server and server-to-client traffic on different ports. This is the default behavior.

pnic apply-firmware

Apply a specific firmware image to the card. You must specify either the firmware name or the complete path of the firmware.

Syntax

pnic apply-firmware [*number*]

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 42 pnic apply-firmware Command Example 1

```
[root@localhost SW]# pnic apply-firmware
No card number specified. Assuming card 0

Do you really want to apply a new firmware for card0 (y/n)? y

Please enter the path or name of the firmware to apply: /usr/local/
pnic/firmware/null.xc4vlx200-ff1513.50.50.2048

Compiling dynamic rules for pnic0

Parsing the dynamic rules for channel0


Parsing the dynamic rules for channel1



Interface pnic0 is down


Waiting for matching to stop ...

Loading rule firmwares ........... Done.

Loading pass/block settings ... Done.

Loading dynamic rules ... Done.

***************************************
Interface pnic0 is up
MTU set to 9264 bytes
***************************************


  Version : P_MAIN2.2.0.058


The firmware image null.xc4vlx200-ff1513.50.50.2048 was successfully
applied to card0

[root@localhost SW]#
```

Figure 43 pnic apply-firmware Command Example 2

```
[root@localhost SW]# pnic apply-firmware
No card number specified. Assuming card 0

Do you really want to apply a new firmware for card0 (y/n)? n

[root@localhost SW]#
```

Related Commands

pnic show-firmwares	Display the available firmware.

pnic capture-off

Disable the capturing of packets via direct memory access (DMA).

Syntax

pnic capture-off

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 44 pnic capture-off Command Example

```
[root@localhost SW]# pnic capture-off
No card number specified. Assuming card 0

Capture OFF set successful.

[root@localhost SW]#
```

Usage Information

Turning off capturing might be desirable during traffic mirroring or pure filtering applications where the host is only used for control.

Related Commands

pnic capture-on	Enable the capturing of packets via direct memory access (DMA).

pnic capture-on

Enable the capturing of packets via direct memory access (DMA).

Syntax

pnic capture-on

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 45 pnic capture-on Command Example

```
[root@localhost SW]# pnic capture-on
No card number specified. Assuming card 0

Capture ON set successful.

[root@localhost SW]#
```

Related Commands

pnic capture-off	Disable the capturing of packets via direct memory access (DMA).

pnic cardstatus

Display the status of the ports, the revision number of the PCI-X FPGA, and the revision number of the Master FPGA.

Syntax

pnic cardstatus [*number*]

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 46 pnic cardstatus Command Example

```
[root@localhost SW]# pnic cardstatus
No card number specified. Assuming card 0

************************************************
----- Channel Port Connection Status -----
Card 0, Channel 0: UP, Active, RX/TX
Card 0, Channel 1: UP, Active, RX/TX
----- Mirror Port Connection Status -----
Card 0, Mirror Port 0: No Carrier
Card 0, Mirror Port 1: No Carrier
************************************************
PCI FPGA revision: 2.8
************************************************
Master FPGA is loaded, revision: 2.6
************************************************

  Version : P_MAIN2.2.0.058

[root@localhost SW]#
```

Related Commands

pnic showconf	Display the configuration parameters of the system.
pnic version	Display the driver version.

pnic compilerules

Transform the dynamic Snort rules contained in */usr/local/pnic/0/rules.custom* into binary code suitable for the DPI processor.

Syntax

pnic compilerules [*number*]

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 47 pnic compilerules Command Example

```
[root@localhost SW]# pnic compilerules
No card number specified. Assuming card 0

Compiling dynamic rules for pnic0

Parsing the dynamic rules for channel0


Parsing the dynamic rules for channel1


  Version : P_MAIN2.2.0.058

[root@localhost SW]#
```

Usage Information

The binary code created by this command is stored in the file */usr/local/pnic/0/pnic_{0|1}.bin*. This command also updates the rule description databases */usr/local/pnic/0/pnic_{0|1}.custmapping*.

pnic default-drop-disable

Disable firewall functionality. This is the default behavior.

pnic default-drop-disable [*number*]

Enable firewall functionality using the command **pnic default-drop-enable**.

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.2.0.0	Introduced

Example

Figure 48 pnic default-drop-disable Command Example

```
[root@localhost SW]# pnic default-drop-disable
No card number specified. Assuming card 0

*** Disabling Default-Packet-Drop on card:0 successful!
*** Temporary memory enabled.
*** Flow teardown disabled.

[root@localhost SW]#
```

pnic default-drop-enable

Enable firewall functionality.

pnic default-drop-enable [*number*]

Disable firewall functionality using the command **pnic default-drop-disable**.

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.2.0.0	Introduced

Example

Figure 49 pnic default-drop-enable Command Example

```
[root@localhost SW]# pnic default-drop-enable
No card number specified. Assuming card 0

*** Enabling Default-Packet-Drop on card:0 successful.
*** Temporary memory disabled.
*** Flow teardown enabled.
[root@localhost SW]#
```

Usage Information

Temporary memory is disabled while the firewall is enabled.

pnic diag

Run diagnostic tests on the card.

Syntax

pnic diag [*number*] [**-v**]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0
-v	Display a detailed output.

Command History

Version 2.3.1.2	Added option **-v**.
Version 2.0.0.1	Introduced

Example

Figure 50 pnic diag Command Example 1

```
[root@localhost pnic]# pnic diag
No card number specified. Assuming card 0

Running PNIC diagnostic test needs to stop traffic matching.
Do you want to proceed [n/y]? y
***  Matching disabled. Test starting ...

Waiting for matching to stop ...

PNIC card 0 is detected on PCI bus.
Software driver module is loaded.

Loading Null firmware ...
Null firmware loading is done

Parsing the dynamic rules for channel0

R=8 alert on c0 ip any any -> any any (msg:"non-ipv4"; )


Parsing the dynamic rules for channel1

R=8 alert on c1 ip any any -> any any (msg:"non-ipv4"; )




Loading rule firmwares ............ Done.

Loading pass/block settings ... Done.

Loading dynamic rules ... Done.
Please run 'pnic restart' or reboot the box to make it operate
normally.

  Version : P_MAIN2.3.0.014

[root@localhost SW]#
```

Example

Figure 51 pnic diag Command Example 2

```
[root@localhost SW]# pnic diag
No card number specified. Assuming card 0

    Running PNIC diagnostic test needs to stop traffic matching.
    Do you want to proceed [n/y]? n
  *** Exit (Diagnostic test aborted). ***
[root@localhost SW]#
```

Usage Information

This CLI provides the ability to diagnose the hardware problems which might appear in registers, memories, or other devices. It reads and writes the registers on the master and PCI FPGAs, which include all configuration registers, counters, MDIO, and PHY registers. It also tests the pass/block setting rule CAM registers. The RAM BIST and initialization are also done in this test.

pnic flow-teardown-disable

Configure the appliance to reset the state of the flow only upon a timeout. This is the default behavior.

Syntax

pnic flow-teardown-disable

Command History

Version 2.3.1.2	Introduced

Example

Figure 52 pnic flow-teardown-disable Command Example

```
[root@localhost SW]# pnic flow-teardown-disable
No card number specified. Assuming card 0


*** Disabling Flow-Teardown on card:0 successful.


[root@localhost SW]#
```

Usage Information

The flow teardown feature is coupled with the firewall feature. When default drop mode is enabled (command **pnic default-drop-enable**), the flow teardown is enabled by default. When default drop mode is disabled (**pnic default-drop-disable**), the flow teardown is disabled by default.

Related Commands

pnic default-drop-disable	Disable firewall functionality. This is the default behavior.
pnic default-drop-enable	Enable firewall functionality.

pnic flow-teardown-enable

Configure the appliance to clear any existing state for a TCP connection in the state memory when it receives a TCP packet with FIN and/or RST bit set.

Syntax

pnic flow-teardown-enable

Command History

Version 2.3.1.2	Introduced

Example

Figure 53 pnic flow-teardown-enable Command Example

```
[root@localhost SW]# pnic flow-teardown-enable
No card number specified. Assuming card 0


*** Enabling Flow-Teardown on card:0 successful.


[root@localhost SW]#
```

Usage Information

The flow teardown feature is coupled with the firewall feature. When default drop mode is enabled (command **pnic default-drop-enable**), the flow teardown is enabled by default. When default drop mode is disabled (**pnic default-drop-disable**), the flow teardown is disabled by default.

Related Commands

pnic default-drop-disable	Disable firewall functionality. This is the default behavior.
pnic default-drop-enable	Enable firewall functionality.

pnic getmachashindex

Display the hash index value for a specific source and destination IP address combination.

Syntax

pnic getmachashindex [*number*]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.1.0.0	Introduced

Example

Figure 54 pnic getmachashindex Command Example

```
[root@localhost SW]# pnic getmachashindex
No card number specified. Assuming card 0

Please input the Source IP address [e.g. 192.168.15.22]: 10.14.122.21
Input the Destination IP address [e.g. 172.168.15.14]: 154.12.123.44

The hash index calculated for MAC rewrite is: 170 (0xaa)

[root@localhost SW]#
```

Usage Information

Use this command with the MAC rewrite feature. This command displays the hash index value for an IP address pairs.

Related Commands

pnic macrewrite-on	Enable MAC rewriting.
pnic macrewrite-off	Disable MAC rewriting.
pnic updatemacvalue	Update the LSB value for a particular hash index value.

pnic gui

Launch the graphical user interface.

Syntax

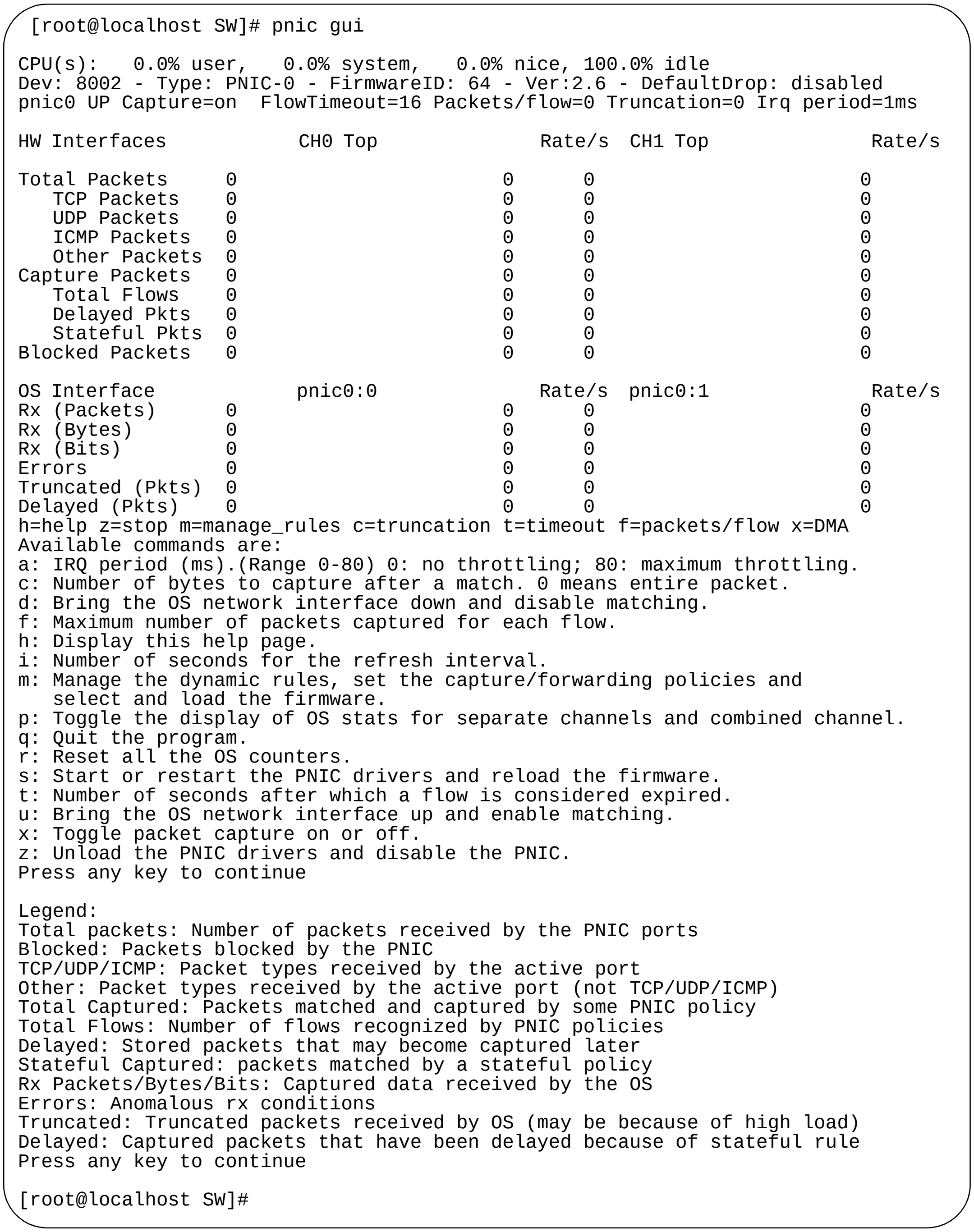
```
pnic gui
```

Command History

Version 2.0.0.1	Introduced

Example

Figure 55 pnic gui Command Example



```
 [root@localhost SW]# pnic gui

CPU(s):   0.0% user,   0.0% system,   0.0% nice, 100.0% idle
Dev: 8002 - Type: PNIC-0 - FirmwareID: 64 - Ver:2.6 - DefaultDrop: disabled
pnic0 UP Capture=on  FlowTimeout=16 Packets/flow=0 Truncation=0 Irq period=1ms

HW Interfaces           CH0 Top              Rate/s  CH1 Top              Rate/s

Total Packets     0                       0      0                       0
   TCP Packets    0                       0      0                       0
   UDP Packets    0                       0      0                       0
   ICMP Packets   0                       0      0                       0
   Other Packets  0                       0      0                       0
Capture Packets   0                       0      0                       0
   Total Flows    0                       0      0                       0
   Delayed Pkts   0                       0      0                       0
   Stateful Pkts  0                       0      0                       0
Blocked Packets   0                       0      0                       0

OS Interface            pnic0:0              Rate/s  pnic0:1              Rate/s
Rx (Packets)      0                       0      0                       0
Rx (Bytes)        0                       0      0                       0
Rx (Bits)         0                       0      0                       0
Errors            0                       0      0                       0
Truncated (Pkts)  0                       0      0                       0
Delayed (Pkts)    0                       0      0                       0
h=help z=stop m=manage_rules c=truncation t=timeout f=packets/flow x=DMA
Available commands are:
a: IRQ period (ms).(Range 0-80) 0: no throttling; 80: maximum throttling.
c: Number of bytes to capture after a match. 0 means entire packet.
d: Bring the OS network interface down and disable matching.
f: Maximum number of packets captured for each flow.
h: Display this help page.
i: Number of seconds for the refresh interval.
m: Manage the dynamic rules, set the capture/forwarding policies and
   select and load the firmware.
p: Toggle the display of OS stats for separate channels and combined channel.
q: Quit the program.
r: Reset all the OS counters.
s: Start or restart the PNIC drivers and reload the firmware.
t: Number of seconds after which a flow is considered expired.
u: Bring the OS network interface up and enable matching.
x: Toggle packet capture on or off.
z: Unload the PNIC drivers and disable the PNIC.
Press any key to continue

Legend:
Total packets: Number of packets received by the PNIC ports
Blocked: Packets blocked by the PNIC
TCP/UDP/ICMP: Packet types received by the active port
Other: Packet types received by the active port (not TCP/UDP/ICMP)
Total Captured: Packets matched and captured by some PNIC policy
Total Flows: Number of flows recognized by PNIC policies
Delayed: Stored packets that may become captured later
Stateful Captured: packets matched by a stateful policy
Rx Packets/Bytes/Bits: Captured data received by the OS
Errors: Anomalous rx conditions
Truncated: Truncated packets received by OS (may be because of high load)
Delayed: Captured packets that have been delayed because of stateful rule
Press any key to continue

[root@localhost SW]#
```

pnic help

Display a list of all available commands, their syntax, and descriptions.

Syntax **pnic help**

Command History

Version 2.3.0.0	Introduced

Example

Figure 56 pnic help Command Example

```
[root@localhost SW]# pnic help
No card number specified. Assuming card 0

Usage: pnic function_command <card_num> <channel_num> <force_options>

pnic aggregate-mode-disable <0|...|5>   pnic aggregate-mode-enable <0|...|5>
pnic apply-firmwares <0|...|5> <-f>     pnic capture-off <0|...|5>
pnic capture-on <0|...|5>               pnic cardstatus <0|...|5>
pnic compilerules <0|...|5>             pnic default-drop-disable <0|...|5>
pnic default-drop-enable <0|...|5>      pnic diag <0|...|5>
pnic getmachashindex <0|...|5>          pnic gui <0|...|5>
pnic linkdown <0|...|5> <0/1>           pnic linkup <0|...|5> <0/1>
pnic loadconf <0|...|5>                 pnic loadeproms <0|...|5>
pnic loadpassblock <0|...|5>            pnic loadrules <0|...|5>
pnic macrewrite-off <0|...|5> <0/1>     pnic macrewrite-on <0|...|5> <0/1>
pnic params <0|...|5>                   pnic passive-mode-enable <0|...|5>
pnic passive-mode-disable <0|...|5>     pnic restart <0|...|5> <-f>
pnic showconf <0|...|5>                 pnic show-firmwares
pnic sguil-sensor-start                 pnic sguil-sensor-stop
pnic start <0|...|5>                    pnic stop <0|...|5>
pnic temp-mem-disable <0|...|5>         pnic temp-mem-enable <0|...|5>
pnic updatemacvalue <0|...|5>           pnic version
pnic web-gui-start                      pnic web-gui-stop
pnic help

Note:
<>             : Option. Default (blank) values are "0"
<card_num>     : Select from 0, 1, 2, 3, 4, or 5
<channel_num>  : Select from 0, or 1
<force_option> : This option will skip the firmware revision check


Command Help:

aggregate-mode-enable    Map both client-to-server and server-to-client traffic
on a channel
                          to the same flow state entry.
aggregate-mode-disable   Map client-to-server and server-to-client traffic from
separate
                          channels to different flow state entries.
apply-firmwares           Apply the selected firmware to the link that is
currently in use or
                          for loading
capture-off(off)          Disable the capture of the packets via DMA.
capture-on(on)            Enable the capture of the packets via DMA.
cardstatus                Display the status of the ports, the revision number
of the PCI-X
                          FPGA, and the revision number of the Master FPGA.
compilerules              Transform the dynamic Snort rules contained in /usr/
local/pnic/0/
                          rules.custom into binary code suitable for the DPI
processor.
default-drop-enable       Enable firewall functionality.
default-drop-disable      Disable firewall functionality.
diag                      Run diagnostic tests on the card.
getmachashindex          Display the hash index value for a specific source and
destination
                          IP address combination.
linkup                    Enable the physical link.
[output omitted]
```

pnic linkdown

Disable the physical link.

Syntax

pnic linkdown [*number*] [*channel*]

Enable a physical link using the command **pnic linkup**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0
channel	Enter the channel number Range: 0-1 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 57 pnic linkdown Command Example

```
[root@localhost SW]# pnic linkdown
No card number specified. Assuming card 0

No channel number specified. Assuming channel 0

Card 0, Channel 0 is down.

[root@localhost SW]#
```

Related Commands

pnic linkup	Enable the physical link ports.

pnic linkup

Enable the physical link.

Syntax

pnic linkup [*number*] [*channel*]

Disable a physical link using the command **pnic linkdown**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0
channel	Enter the channel number Range: 0-1 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 58 pnic linkup Command Example

```
[root@localhost SW]# pnic linkup
No card number specified. Assuming card 0

No channel number specified. Assuming channel 0

Card 0, Channel 0 is up.

[root@localhost SW]#
```

Related Commands

pnic linkdown	Enable the physical link ports.

pnic loadconf

Upload the runtime configuration parameters contained in the file */usr/local/pnic/0/pnic.conf*.

Syntax

pnic loadconf [*number*]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 59 pnic loadconf Command Example

```
[root@localhost ~]# pnic loadconf
No card number specified. Assuming card 0

Loading configurations ...
Read from configuration file and apply to PNIC card...
Registers on master FPGA:
(0x10)0000 (0x14)0010 (0x18)0000
Registers on PCI FPGA:
(0x18)0100 (0x24)20788 (0x28)20788

DMA Capture               : on
MAC rewrite              : CH0 - disabled; CH1 - disabled
Default Drop packet       : disabled
Temporary memory          : enabled
Aggregate mode            : enabled
Flow teardown             : disabled
PHY passive mode          : disabled
Vlan remove              : disabled

Read out the registers that were just applied.
On MASTER FPGA
(0x10)00000000 (0x14)00000010 (0x18)00000000
On PCI FPGA
(0x18)00000100 (0x24)00020788 (0x28)00020788

DMA Capture                         : on
MAC rewrite                         : CH0 - disabled; CH1 - disabled
Default Drop packet               : disabled
Temporary memory                : enabled
Aggregate mode                     : enabled
PHY passive mode                 : disabled
Flow teardown                       : disabled
Vlan remove                         : disabled

Version : P_PRIV2.3.0.010
```

Usage Information

The syntax of such parameter files is (*address*) *value* where *address* is the decimal address of the DPI control register, and *value* is the hexadecimal parameter to be loaded. Table 27 shows the parameters to which each address is mapped.

Table 26 pnic loadconf Address Mapping

Address	Corresponding Parameter
Address 20 (Master FPGA)	This address is mapped to the parameter *Flow timeout* (measured in multiples of 0.86 seconds). This parameter controls how quickly the stateful packet analysis can garbage-collect previous states. Smaller values increase the number of concurrent flows that can be tracked. The default value is 16.
Address 16 (Master FPGA)	This address is mapped to the parameter *Flow length* (measured in packets). This parameter controls the maximum number of packets in a flow that are considered for capturing. Typical values range from 6 to16.
Address 24 (PCI-X FPGA)	This address is mapped to the parameter *Burst size* (measured in 32-bit words). This parameter sets the number of 32-bit words to transfer in one PCI-X master cycle. Larger bursts achieve higher throughput but may increase buffering latency and contention with other devices sharing the same bus. The default value is 1024.
Address 36 (PCI-X FPGA)	This address specifies the count in PCI-X clocks before the DMA buffer is transferred to the host if the buffer contains less than the programmed burst size.

pnic loadeproms

Load the PCI-X and front-end EEPROMs.

Syntax **pnic loadeproms** [*number*]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Usage Information Use this command to upgrade PCI-X and front-end EEPROMs to new revisions. Reboot the chassis after executing this command; only then does new firmware take effect.

➡ **Note:** This process takes up to 30 minutes.

pnic loadparams (deprecated)

Upload the runtime configuration parameters contained in the file */usr/local/pnic/0/pnic.conf*.

Syntax **pnic loadparams** [*number*]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 60 pnic loadparams Command Example

```
[root@localhost ~]# pnic loadparams
No card number specified. Assuming card 0

Loading configurations...
Read from configuration file and apply to PNIC card...
(0x10)0000 (0x14)0010 (0x18)0000
(0x18)0100 (0x24)20788 (0x28)20788
DMA Capture Status: off
MAC Rewrite state: CH0 - disabled; CH1 - disabled
Default Drop Packet: disabled
Temporary memory: disabled
Aggregate mode: enabled
Passive mode: disabled

Read out the registers that were just applied.
On MASTER FPGA
(0x10)00000000 (0x14)00000010 (0x18)00000000
On PCI FPGA
(0x18)00000100 (0x24)00020788 (0x28)00020788

DMA Capture                         : off

MAC Rewrite state                   : CH0 - disabled; CH1 - disabled

Default Drop Packet                 : disabled

Temporary memory                    : enabled

Aggregate mode                      : enabled

PHY passive mode                    : disabled



  Version : P_MAIN2.2.0.062

[root@localhost ~]#
```

Usage Information

The syntax of such parameter files is (*address*) *value* where *address* is the decimal address of the DPI control register, and *value* is the hexadecimal parameter to be loaded. Table 27 shows the parameters to which each address is mapped.

Table 27 Loadparams Address Mapping

Address	Corresponding Parameter
Address 20 (Master FPGA)	This address is mapped to the parameter *Flow timeout* (measured in multiples of 0.86 seconds). This parameter controls how quickly the stateful packet analysis can garbage-collect previous states. Smaller values increase the number of concurrent flows that can be tracked. The default value is 16.
Address 16 (Master FPGA)	This address is mapped to the parameter *Flow length* (measured in packets). This parameter controls the maximum number of packets in a flow that are considered for capturing. Typical values range from 6 to16.

Table 27 Loadparams Address Mapping

Address	Corresponding Parameter
Address 24 (PCI-X FPGA)	This address is mapped to the parameter *Burst size* (measured in 32-bit words). This parameter sets the number of 32-bit words to transfer in one PCI-X master cycle. Larger bursts achieve higher throughput but may increase buffering latency and contention with other devices sharing the same bus. The default value is 1024.
Address 36 (PCI-X FPGA)	This address specifies the count in PCI-X clocks before the DMA buffer is transferred to the host if the buffer contains less than the programmed burst size.

pnic loadrules

Upload to the FPGA the dynamic rules for both channels encoded in the files */usr/local/pnic/0/pnic_{0|1}.bin*.

Syntax

pnic loadrules [*channel*]

Parameters

channel	Enter the channel number Range: 0-1 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 61 pnic loadrules Command Example

```
root@# pnic loadrules 0
dynamic rules loaded
```

Usage Information

Capture/block policies previously stored are temporarily disabled during this operation and traffic is forwarded. The new rules take effect when the loading process is complete.

pnic macrewrite-off

Disable MAC rewriting. This is the default behavior.

Syntax

pnic macrewrite-off [*number*] [*channel*]

Enable MAC rewriting using the command **pnic macrewrite-on**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0
channel	Enter the channel number Range: 0-1 Default: 0

Command History

Version 2.1.0.0	Introduced

Example

Figure 62 pnic macrewrite-off Command Example

```
[root@localhost SW]# pnic macrewrite-off
No card number specified. Assuming card 0

No channel number specified. Assuming channel 0
*** Disabling MAC rewrite on card:0 channel:0 successful.

[root@localhost SW]#
```

Usage Information

MAC rewriting can be used for load balancing. Load balancing is achieved by overwriting the least significant byte of the destination MAC address for packets with a specified source and destination IP address with a user specified value.

Related Commands

pnic macrewrite-on	Rewrite the least significant byte (LSB) of the destination MAC address for packets with particular source and destination IP addresses.

pnic macrewrite-on

Rewrite the least significant byte (LSB) of the destination MAC address for packets with particular source and destination IP addresses.

Syntax

pnic macrewrite-on [*number*] [*channel*]

Disable MAC rewriting using the command **pnic macrewrite-off**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0
channel	Enter the channel number Range: 0-1 Default: 0

Default

MAC rewrite is disabled by default. The default value for the LSB is the system-assigned hash index value.

Command History

Version 2.1.0.0	Introduced

Example

Figure 63 pnic macrewrite-on Command Example

```
[root@localhost SW]# pnic macrewrite-on
No card number specified. Assuming card 0

No channel number specified. Assuming channel 0
*** Enabling MAC rewrite on card:0 channel:0 successful.

[root@localhost SW]#
```

Usage Information

MAC rewriting can be used for load balancing. Load balancing is achieved by overwriting the least significant byte of the destination MAC address for packets with a specified source and destination IP address with a user specified value.

Related Commands

pnic macrewrite-off	Disable MAC rewriting.

pnic off (deprecated)

Disable the capturing of packets via direct memory access (DMA).

Syntax

pnic off

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 64 pnic off Command Example

```
[root@localhost SW]# pnic off
No card number specified. Assuming card 0

Capture OFF set successful.

[root@localhost SW]#
```

Usage Information

Turning off capturing might be desirable during traffic mirroring or pure filtering applications where the host is only used for control.

Related Commands

pnic on (deprecated)	Enable the capturing of packets via direct memory access (DMA).

pnic on (deprecated)

Enable the capturing of packets via direct memory access (DMA).

Syntax

pnic on

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 65 pnic on Command Example

```
[root@localhost SW]# pnic on
No card number specified. Assuming card 0

Capture ON set successful.

[root@localhost SW]#
```

Related Commands

pnic off (deprecated)	Disable the capturing of packets via direct memory access (DMA).

pnic params

Display the card interface name, device ID, and contents of the register on the PCI-X and Master FPGAs.

Syntax

pnic params [*number*]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 66 pnic params Command Example

```
[root@localhost SW]# pnic params
No card number specified. Assuming card 0



PNIC 8002 pnic0 0xffff810000700000 20006

**********************  Register Display  **********************
**** Configurations on Master FPGA *****************************
Register Name                           (Address) Hex
----------------------------------------------------------------
Revision                                (0x000)80020006
Chip Control                            (0x004)00000073
Scratch                                 (0x008)75318642
Chip Status                             (0x00c)00000003
Packet Linked List Limit                (0x010)00000000
Timeout for Flow Garbage Collection     (0x014)00000010
Byte Number of Truncation with Match    (0x018)00000000
Time Stamp for Sync                     (0x01c)4C787C4B
RAM Failure Address                     (0x020)00000000
RAM Failure Data Low                    (0x024)00000000
RAM Failure Data Meddle                 (0x028)00000000
RAM Failure Data High                   (0x02c)00000000
[output omitted]
```

pnic passive-mode-disable

Configure the ports to transmit and receive traffic. This is the default behavior.

Syntax

pnic passive-mode-disable [*number*]

Enable passive mode using the command **pnic passive-mode-enable**.

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 67 pnic passive-mode-disable Command Example

```
[root@localhost SW]# pnic passive-mode-disable
No card number specified. Assuming card 0

Channel 0 and 1 are set to work in normal TX/RX mode.

[root@localhost SW]#
```

Related Commands

pnic passive-mode-enable	Configure the ports to only receive traffic.

pnic passive-mode-enable

Configure the ports to only receive traffic.

Syntax

pnic passive-mode-enable [*number*]

Disable passive mode using the command **pnic passive-mode-disable**.

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 68 pnic passive-mode-enable Command Example

```
[root@localhost SW]# pnic passive-mode-enable
No card number specified. Assuming card 0

Channel 0 and 1 are set to work in passive mode.

[root@localhost SW]#
```

Related Commands

pnic passive-mode-disable	Receive both client-to-server and server-to-client traffic on one port.

pnic resetconf

Reset the system configuration back to the default settings, which are located in *<installation_directory>/SW/misc/pnic.conf*.

Syntax

pnic resetconf [*number*]

Parameters

number	(OPTIONAL) Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.1.2	Introduced

Example

Figure 69 pnic resetconf Command Example

```
[root@localhost ~]# pnic resetconf
No card number specified. Assuming card 0

Loading default configurations ...
Read from configuration file and apply to PNIC card...
Registers on master FPGA:
(0x10)0000 (0x14)0010 (0x18)0000
Registers on PCI FPGA:
(0x18)0100 (0x24)20788 (0x28)20788

DMA Capture               : on
MAC rewrite               : CH0 - disabled; CH1 - disabled
Default Drop packet       : disabled
Temporary memory          : enabled
Aggregate mode            : enabled
Flow teardown             : disabled
PHY passive mode          : disabled
Vlan remove               : disabled

Read out the registers that were just applied.
On MASTER FPGA
(0x10)00000000 (0x14)00000010 (0x18)00000000
On PCI FPGA
(0x18)00000100 (0x24)00020788 (0x28)00020788

DMA Capture                          : on
MAC rewrite                          : CH0 - disabled; CH1 - disabled
Default Drop packet                  : disabled
Temporary memory                     : enabled
Aggregate mode                       : enabled
PHY passive mode                     : disabled
Flow teardown                        : disabled
Vlan remove                          : disabled

  Version : P_MAIN2.3.0.006

[root@localhost ~]#
```

pnic restart

- Stop capturing and matching

- Load the rule firmware
- Load the capture/block configuration
- Load the runtime parameters
- Enable the network interface

Note: Essentially, this command performs the command **pnic stop** followed by the command **pnic start**.

Syntax

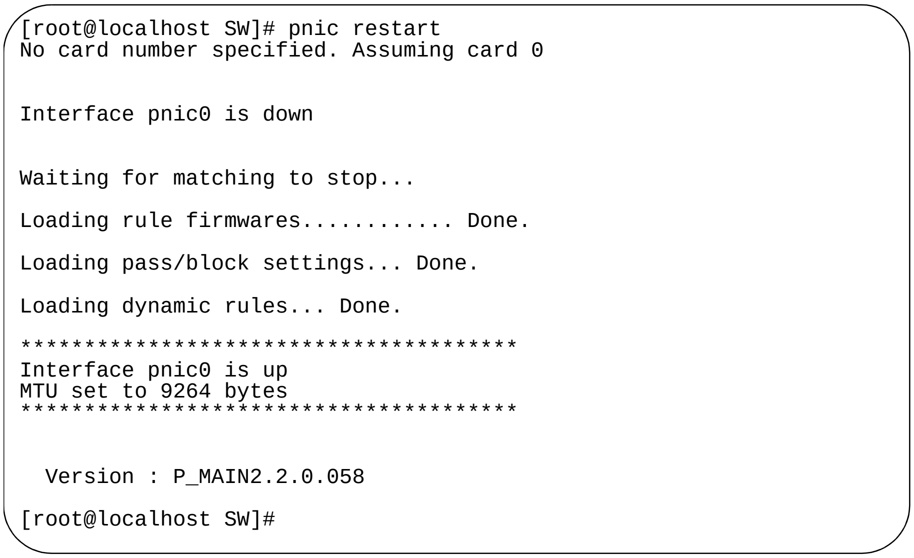
pnic restart

Command History

Version 2.0.0.1	Introduced

Example

Figure 70 pnic restart Command Example



```
[root@localhost SW]# pnic restart
No card number specified. Assuming card 0


Interface pnic0 is down


Waiting for matching to stop...

Loading rule firmwares........... Done.

Loading pass/block settings... Done.

Loading dynamic rules... Done.

***************************************
Interface pnic0 is up
MTU set to 9264 bytes
***************************************


  Version : P_MAIN2.2.0.058

[root@localhost SW]#
```

Usage Information

restart always reloads the FPGA, as opposed to **start** which does not load the FPGA if firmware is already present.

Related Commands

pnic stop	Disable the network interface.
pnic start	Enable the network interface.

pnic sguil-sensor-start

Start the Sguil sensor.

Syntax

pnic sguil-sensor-start [**-f**]

Stop the Sguil sensor using the command **pnic sguil-sensor-stop**.

Parameters

-f	The first time the sensor starts, the you are prompted for parameters. Those parameters are stored in configuration files and reused. Specify this option to be prompted for new parameter values.

Command History

Version 2.3.0.0	Introduced

Example

Figure 71 pnic sguil-sensor-start Command Example

```
[root@localhost pnic]# pnic sguil-sensor-start


Enter the IP address of the Sguil-Server:10.11.194.183

Do you want to enable secure connection between sguil-sensor and
sguil-server?
1) Enable
2) Disable
#? 1

***********************************************
INTERFACE NAME          : pnic0
SGUIL-SERVER IP-ADDRESS : 10.11.194.183
SECURE CONNECTIVITY     : Enabled
***********************************************

To start Sguil-sensor with the above configuration
Select "Ok"

1) Ok
2) Exit
#? 1
Starting sguil sensor processes...
Starting barnyard...
Starting snort...
Snort is already running
Starting sancp...
Sancp is already running
Starting new process...LogPackets started successfully.
Killing old process...Old LogPackets process killed successfully.
Checking disk space (limited to 90%)...
  Current Disk Use: 19%
Done.
Starting Pcap Agent...
Pcap Agent already running
Starting Sancp Agent...
Sancp Agent already running
Starting Snort Agent...
Snort Agent already running
Barnyard started successfully.
Snort started successfully.
Sancp started successfully.
Pcap Agent started successfully.
Sancp Agent started successfully.
Snort Agent started successfully.
Sguil-sensor has started successfully.

[root@localhost pnic]#
```

Related Commands

pnic sguil-sensor-stop	Stop the Sguil sensor.

pnic sguil-sensor-stop

Stop the Sguil sensor.

Syntax

pnic sguil-sensor-stop [**-f**]

Start the Sguil sensor using the command **pnic sguil-sensor-start**.

Parameters

-f	Exit the Squil sensor without a confirmation prompt.

Command History

Version 2.3.0.0	Introduced

Example

Figure 72 pnic sguil-sensor-stop Command Example 1

```
[root@localhost pnic]# pnic sguil-sensor-stop

Do you really want to stop the Sguil-sensor application (y/n)? y

LogPackets stopped successfully.
Trying to stop Pcap Agent
Stopped Pcap Agent successfully
Trying to stop Sancp Agent
Stopped Sancp Agent successfully
Trying to stop Snort Agent
Stopped Snort Agent successfully
Trying to stop Barnyard
Barnyard is not running
Trying to stop Snort
Stopped Snort successfully
Trying to stop Sancp
Stopped Sancp successfully
Trying to stop tail of snort.stats started by sensor_agent
Stopped tail of snort.stats successfully
The Sguil-sensor application has been stopped!

[root@localhost pnic]#
```

Figure 73 pnic sguil-sensor-stop Command Example 2

```
[root@localhost SW]# pnic sguil-sensor-stop

Do you really want to stop the Sguil-sensor application (y/n)? n

[root@localhost SW]#
```

Related Commands

pnic sguil-sensor-start	Start the Sguil sensor.

pnic showconf

Display configuration parameters of the card.

Syntax **pnic showconf** [*number*]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example **Figure 74** pnic showconf Command Example

```
[root@localhost ~]# pnic showconf
No card number specified. Assuming card 0

DMA Capture                         : on
MAC rewrite                         : CH0 - disabled; CH1 - disabled
Default Drop packet              : disabled
Temporary memory                : enabled
Aggregate mode                     : enabled
PHY passive mode                 : disabled
Flow teardown                       : disabled
Vlan remove                         : disabled

####################### On MASTER FPGA #######################

Per Flow Packet Limit           : unlimited
Timeout for Flow Garbage Collection : 16
Truncation after Match Packet       : full packet

####################### On PCI FPGA #######################

DMA Burst Size                    : 1024 (Bytes)
DMA Flush Timer                 : 1 (ms)
Interrupt Frequency Timer    : 1 (ms)

  Version : P_PRIV2.3.0.010
```

Related Commands

pnic cardstatus	Display the status of the ports, the revision number of PCI-X FPGA, and the revision number of the Master FPGA.
pnic version	Display the driver version.

pnic show-firmwares

List the available firmware images.

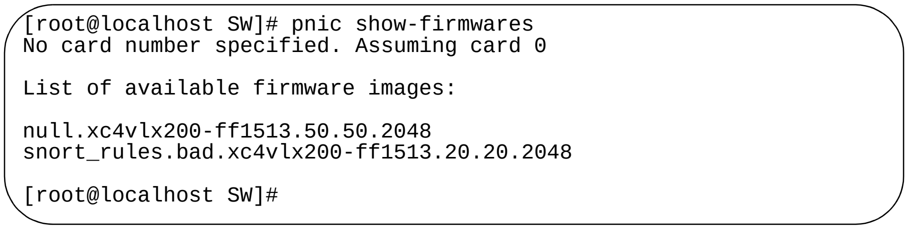
Syntax **pnic show-firmwares**

Command History

Version 2.3.0.0	Introduced

Example

Figure 75 pnic show-firmwares Command Example



```
[root@localhost SW]# pnic show-firmwares
No card number specified. Assuming card 0

List of available firmware images:

null.xc4vlx200-ff1513.50.50.2048
snort_rules.bad.xc4vlx200-ff1513.20.20.2048

[root@localhost SW]#
```

Related Commands

pnic apply-firmware	Apply a specific firmware to the card.

pnic showtech

Display all technical data and configuration files for the diagnostic and debugging purpose.

Syntax

pnic showtech [*number*] [**>***filename***.dat**]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0
filename	Save the output to a file.

Command History

Version 2.3.1.2	Introduced

Example

Figure 76 pnic showtech Command Example

```
[root@localhost pnic]# pnic showtech | more
No card number specified. Assuming card 0


************************************************************
      Display date
************************************************************

Tue Apr 29 11:21:07 PDT 2008

************************************************************
      Display OS version information
************************************************************

Linux localhost.localdomain 2.6.18-8.1.14.el5 #1 SMP Thu Sep 27
19:05:32 EDT 2007 x86_64 x86_64 x86_64 GNU/Linux
CentOS release 5 (Final)

************************************************************
      Display CPU usage
************************************************************



top - 11:21:08 up 23:01,  3 users,  load average: 0.03, 0.02, 0.00
Tasks:  76 total,   1 running,  75 s
leeping,   0 stopped,   0 zombie
Cpu(s):  0.6%us,  0.5%sy,  0.0%ni,
 98.8%id,  0.2%wa,  0.0%hi,
[output omitted]
```

pnic start

- Load the rule firmware if it is not already present
- Load the capture/block configuration
- Load the runtime parameters
- Enable the network interface.

Syntax

pnic start [*number*]

Disable the network interface using the command **pnic stop**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 77 pnic start Command Example

```
[root@localhost SW]# pnic start
No card number specified. Assuming card 0


Interface pnic0 is down

Loading pass/block settings ... Done.

Loading dynamic rules ... Done.

***************************************
Interface pnic0 is up
MTU set to 9264 bytes
***************************************


  Version : P_MAIN2.2.0.058

[root@localhost SW]#
```

Related Commands

pnic stop	Disable the network interface.

pnic stop

Turn off capture and disable the network interface.

Syntax

pnic stop [*number*]

Enable the network interface using the command **pnic start**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.0.0.1	Introduced

Example

Figure 78 pnic stop Command Example

```
[root@localhost SW]# pnic stop
No card number specified. Assuming card 0

Interface pnic0 is down
[root@localhost SW]#
```

Related Commands

pnic start	Enable the network interface.

Usage Information

This command disables the pnic software interface and disables capturin to the CPU, but the card still forwards/blocks traffic on the wire.

pnic temp-mem-disable

Disable temporary memory.

Syntax

pnic temp-mem-disable [*number*]

Enable temporary memory using the command **pnic temp-mem-enable**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 79 pnic temp-mem-disable Command Example

```
[root@localhost SW]# pnic temp-mem-disable
No card number specified. Assuming card 0

*** Disabling temporary memory on card:0 successful.

[root@localhost SW]#
```

Related Commands

pnic temp-mem-enable	Enable temporary memory.

Usage Information

Disabling the temporary memory reduces flow collisions. In firewall mode (default-drop), temporary memory is disabled automatically.

pnic temp-mem-enable

Enable temporary memory. This is the default behavior.

Syntax

pnic temp-mem-enable [*number*]

Disable temporary memory using the command **pnic temp-mem-disable**.

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.3.0.0	Introduced

Example

Figure 80 pnic temp-mem-enable Command Example

```
[root@localhost SW]# pnic temp-mem-enable
No card number specified. Assuming card 0

*** Enabling temporary memory on card:0 successful.

[root@localhost SW]#
```

Related Commands

pnic temp-mem-disable	Disable temporary memory.

pnic updatemacvalue

Specifies an LSB value for a particular hash index.

Syntax

pnic updatemacvalue [*number*]

Parameters

number	Enter the number of the network interface card. Range: 0-5 Default: 0

Command History

Version 2.1.0.0	Introduced

Example

Figure 81 pnic updatemacvalue Command Example

```
[root@localhost SW]# pnic updatemacvalue
No card number specified. Assuming card 0

Please input the hash index [0-255]: 56
The value to replace: 0x78
The MAC address updating is done on register 0x4e0 - index:56

[root@localhost SW]#
```

Related Commands

pnic macrewrite-on	Enable MAC rewriting.
pnic macrewrite-off	Disable MAC rewriting.
pnic updatemacvalue	Obtain or assign a MAC LSB hash index value.

Usage Information

Use this command with the MAC rewrite feature.

pnic vlan-remove-disable

Disable the VLAN Tag Remove feature.

Syntax **pnic vlan-remove-disable**

Default The VLAN Tag Remove feature is *disabled* by default.

Command History

Version 2.3.1.2	Introduced

Usage Information This feature is enabled and disabled on both sensing ports.

Example

Figure 82 pnic vlan-remove-disable Command Example

```
[root@localhost pnic]# pnic vlan-remove-disable
No card number specified. Assuming card 0

*** Disabling VLAN tag remove on card:0 channel 0&1 successful.

[root@localhost pnic]#
```

pnic vlan-remove-enable

Remove the VLAN tag and recalculate the CRC on all tagged packets passing through the appliance.

Syntax **pnic vlan-remove-enable**

Default The VLAN Tag Remove feature is *disabled* by default.

Command History

Version 2.3.1.2	Introduced

Usage Information This feature is enabled and disabled on both sensing ports.

Example

Figure 83 pnic vlan-remove-enable Command Example

```
[root@localhost pnic]# pnic vlan-remove-enable
No card number specified. Assuming card 0

*** Enabling VLAN tag remove on card:0 channel 0&1 successful.

[root@localhost pnic]#
```

pnic version

Display the driver version.

Syntax **pnic version**

Command History

Version 2.0.0.1	Introduced

Example

Figure 84 pnic version Command Example

```
[root@localhost SW]# pnic version

Force10 Networks PNIC Software Version: P_MAIN2.2.0.058

[root@localhost SW]#
```

pnic web-gui-start

Start the web server.

Syntax **pnic web-gui-start** [**-f**]

Disable the web server using the command **pnic web-gui-stop**.

Parameters

-f	The first time the Web server is started, the P10 prompts for and stores parameters to generate a self-signed certificate. From then on, the same certificate is used when starting the server when you enter the command. If you specify the **-f** option, the P10 prompts you again for the parameters to generate a new certificate.

Command History

Version 2.3.0.0	Introduced

Example

Figure 85 pnic web-gui-start Command Example

```
[root@localhost pnic]# pnic web-gui-start


INFO: Generating SSL certificate for the web-gui application.

Generating a 1024 bit RSA private key
.........++++++
......++++++
writing new private key to '/usr/local/pnic-mgmt-lib/sslcert/rootkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (FQDN or IP address of the P-Series box using which you access the
web-gui application) [192.168.1.1]:10.11.194.184
Organization Name (company) [Force10 Networks Inc]:
Organizational Unit Name (department, division) [P-Series Security]:
Locality Name (city, district) [350 Holger way, San Jose]:
State or Province Name (full name) [California]:
Country Name (2 letter code) [US]:
Email Address [support@force10networks.com]:

INFO: SSL certificate generation for the web-gui application successful.

Web-gui application has started successfully!

[root@localhost pnic]#
```

Related Commands

pnic web-gui-stop	Stop the web server.

pnic web-gui-stop

Stop the web server.

Syntax

pnic web-gui-stop [**-f**]

Enable the web server using the command **pnic web-gui-start**.

Parameters

-f	Stop the Web-gui server without a confirmation prompt.

Command History

Version 2.3.0.0	Introduced

Example

Figure 86 pnic web-gui-stop Command Example

```
[root@localhost pnic]# pnic web-gui-stop
Do you really want to stop the web-gui application (y/n)? y
Web-gui application has been stopped!
[root@localhost pnic]#
```

Related Commands

pnic web-gui-start	Start the web server.

Appendix B Snort Keywords

Table 28 describes briefly the valid Snort keywords supported on the P-Series. For a more detailed explanation for these keywords, see the Snort website at http://www.snort.org/docs/snort_manual/node17.html.

Table 28 Description of P-Series Snort Keywords

<table>
<tr><th>Keyword</th><th>Description</th><th>Rule Syntax</th></tr>
<tr><td>ack</td><td>Checks for a specific TCP acknowledgment number.
number is a reference to a previously transmitted sequence number that is being acknowleged.</td><td>**ack:** *number;*</td></tr>
<tr><td>content</td><td>Specifies the content within the packet payload for which the rule is to search.
data_string can contain mixed text and binary data. Binary data is enclosed within pipe characters and is written in hexadecimal form.</td><td>**content:** [**!**] **"***data_string***";**</td></tr>
<tr><td>dsize</td><td>Inspects the packet payload size.
number is the payload size in bytes.</td><td>**dsize:** [**&gt;**|**<**] *number* [**>**|**<***number*]**;**</td></tr>
<tr><td>flags</td><td>Checks for the presence of the specified TCP flag bits. Valid flag bits include:
• **F**: FIN (Least Significant Bit (LSB) in the TCP Flags byte)
• **S**: SYN
• **R**: RST
• **P**: PSH
• **A**: ACK
• **U**: URG
• **1**: Reserved bit 1 (Most Significant Bit (MSB) in TCP Flags byte)
• **2**: Reserved bit 2
• **0**: No TCP Flags Set
The following modifiers change the match criteria:
• **+**: Match on the specified bits, plus any others.
• *****: Match if any of the specified bits are set.
• **!**: Match if the specified bits are not set.</td><td>**flags:**[**!**|*****|**+**] {**F**|**S**|**R**|**P**|**A**|**U**|**1**|**2**|**0**} [,{**F**|**S**|**R**|**P**|**A**|**U**|**1**|**2**|**0**}]**;**</td></tr>
</table>

Table 28 Description of P-Series Snort Keywords

Keyword	Description	Rule Syntax
flow	This keyword applies the rule to a specific traffic flow direction. The flow can be in one of two states: • **established**: Trigger only on established TCP connections. • **stateless**: Trigger regardless of the state of the stream processor. The *direction* parameter has the following options: • **to_client**: Trigger on server responses from A to B. • **to_server**: Trigger on client requests from A to B. • **from_client**: Trigger on client requests from A to B. • **from_server**: Trigger on server responses from A to B. • **no_stream**: Do not trigger on rebuilt stream packets. • **only_stream**: Only trigger on rebuilt stream packets.	**flow:** [**established|stateless**] [, *direction*]**;**
icmp_id	This keyword checks for a specific ICMP ID value.	**icmp id:***number***;**
icmp_seq	This keyword checks for a specific ICMP sequence value.	**icmp seq:** *number***;**
icode	This keyword checks for a specific ICMP code value.	**icode:** [>|<] *number* [{**>|<**} *number*]**;**
id	This keyword checks the IP ID field for the specified value.	**id:***number***;**
ip_proto	This keyword inspects the IP protocol header.	**ip_proto:** [**!**|>|<] {*name* |*number*}**;**
itype	This keyword checks for the specified ICMP type value.	**itype:**[>|<] *number* [{>|<} *number*]**;**
nocase	This keyword matches strings without regard for capitalization. This keyword modifies the content keyword.	**nocase;**
protocol	Enter the protocol.	{**ICMP | UDP | TCP | IP**}
seq	This keyword checks for the specified TCP sequence number.	**seq:***number***;**
source address	Enter the address from which traffic is arriving. The	*A.B.C.D*/{*subnet_mask*}
destination address	Enter the address to which traffic is destined.	*A.B.C.D*/{*subnet_mask*}
souce port	Enter the port from which traffic is arriving.	*port_number*
destination port	Enter the port to which traffic is destined.	*port_number*
tos	This keyword checks for the specified ToS value.	**tos:** [**!**] *number***;**

Table 28 Description of P-Series Snort Keywords

Keyword	Description	Rule Syntax
ttl	This keyword checks for the specified IP time-to-live value.	**ttl:** [*number* {>|<|=} | *number*- | {-|>|<|=}] *number*;
uricontent	Searches the normalized request URI field for the specified content. *data_string* can contain mixed text and binary data. Binary data is enclosed within pipe characters and is written in hexadecimal form.	**uricontent**: [**!**] **"***data_string***";**

Appendix C Meta and Evasion Rules

The meta and evasion rules for Channel 0 and Channel 1 are the same. They are listed in Table 29 and Table 30.

Table 29 meta Rules for Channel 0 and Channel 1

meta Rules
alert tcp any any -> any any (msg:"Z SYN"; flags:S,12; S:1; R:2; C:3;)
alert tcp any any -> any any (msg:"Z SYNACK"; flags:SA; S:1; R:2; C:5;)
alert tcp any any -> any any (msg:"Z TCP within was issued previously for this flow = capture flow"; S:32; R:2; C:32;)
alert udp any any -> any any (msg:"Z UDP within was issued previously for this stream = capture stream"; S:64; R:2; C:64;)
alert tcp any any -> any any (msg:"Z SAPU TCP Flags"; flags:SAPU;)
alert tcp any any -> any any (msg:"Z FU TCP Flags"; flags:FU;)
alert tcp any any -> any any (msg:"Z PF TCP Flags"; flags:PF;)
alert tcp any any -> any any (msg:"Z UP TCP Flags"; flags:UP;)
alert tcp any any -> any any (msg:"Z Zero TCP Flags"; flags:0;)

Table 30 Evasion Rules for Channel 0 and Channel 1

Evasion Rules
alert tcp any any -> any any (msg:"Z Evasion: State 2 Fragment of size 1 "; dsize: 1; S:4; R:1; C:16;)
alert tcp any any -> any any (msg:"Z Evasion: State 1 First fragment of size 0 <> 10 = state 1"; dsize: 0 <> 20; S:4; R:1; C:8;)
alert tcp any any -> any any (msg:"Z Evasion: State 2 Second fragment of size 0 <> 10 = capture flow"; dsize: 0 <> 20; S:8; R:1; C:16;)
alert tcp any any -> any any (msg:"Z Evasion: State 3 Capture flow fragments of size 0 <> 10"; dsize: 0 <> 100; S:16; R:2; C:17;)

Appendix D Basic Unix Commands

Unix Commands

Table 31 Basic Unix Commands

Command	Description
cd *path*	Changes the current directory to the specified directory. The path specified can be an absolute path, or a relative path: • The absolute path begins with a forward slash, and specifies the destination directory beginning from the top of the directory tree. • The relative path does not begin with a forward slash, and specifies the destination beginning from a point common between the current and destination directories.
grep *text filename*	Searches the specified file for a specified string of characters.
logout	Logs you out of the current session.
ls *directory*	Displays the contents of the specified directory.
man *command*	Diplays the online manual pages for the specified command.
mkdir *directory*	Makes a directory in the specified location.
more *filename*	Displays the contents of a file one screenful at a time.
mv *directory target*	Moves the specified directory to the target location.
passwd	Allow you to change the current password.
pwd	Displays the directory in which you are currently (present working directory).
rmdir *directory*	Removes the specifed directory. Two conditions apply to this command: • The specified directory must be empty. • The specified directory must not be between the current directory and root directory.

vi Commands

vi has two modes:

- *Command Mode*: In command mode, commands can be entered which allow you to jump to points in a file, search text, and exit the editor.
- *Insert Mode*: Insert mode allows you to create or alter text in a file.

Note: Commands are case sensitive.

Table 32 Basic *vi* Commands

Command	Description
vi *filename*	Opens the specified file in the editor. If the filename does not exits, *vi* creates it. Enter this command from the Unix shell prompt.
(Escape Key)	Exits Insert Mode and enters Command Mode.
(Arrow Keys)	Moves the cursor up, down, left, and right.
i	Enters Insert Mode and allows you to insert text at the current cursor position.
x	Deletes the character at the current cursor position.
{**/** | **?** } **text**	• The command **/ text** Searches for the specified text in the forward direction. • The command **? text** searches for the specified text in the backwards direction.
[**n** | **1**]**G**	• The command **nG** moves the cursor to the specified line, where *n* is the line number. • The command **1G** moves the cursor to the first line in the file. • The command **G** moves the cursor to the last line in the file.
0	Moves the cursor to the beginning of the current line.
$	Moves the cursor to the end of the current line.
:set {**number** | **no number**}	Turns the line numbers on and off.
:q!	Exits the editor without saving changes.
:wq	Saves changes and exits the editor.

Appendix E Glossary

ACK An Acknowledgment packet (ACK) is a packet that is sent from the client to the server to complete a TCP connection. See SYN.

DHCP Dynamic Host Configuration Protocol (DHCP) is a protocol that automatically requests an IP address, subnet mask, and default gateway for a network client.

DMA Direct Memory Access (DMA) is a method by which devices in a hardware system can transfer data without occupying the CPU. In the case of the P-Series, the network interface card can transfer matched packets directly to the host memory by taking control of the PCI-X bus.

DPI Dynamic Parallel Inspection (DPI) is an engine based on Multiple Instruction Single Data (MISD) hardware architecture that can simultaneously execute thousands of security policies and capture/blocking operations on the same data.

Dynamic Rules Dynamic rules allocate generic registers inside the firmware to allow you to create and modify rules at runtime without changing the firmware.

Flow A flow is a series of packets with the same state. See State.

FPGA Field Programmable Gate Array (FPGA) is a logic device that is re-programmable; it is a counterpart to the Application-Specific Integrated Circuit (ASIC) that cannot be modified once it has been programmed.

Garbage Collection Garbage is data that is no longer necessary; garbage collection is the process of discarding this data to free resources. In the context of the P-Series, garbage is old state or flows.

IDS/IPS Intrusion Detection System/Intrusion Prevention System

MISD Multiple Instruction Single Data (MISD) is a computer architecture that executes many operations simultaneously on one set of data. It is a counterpart to Single Instruction Multiple Data (SIMD) and Multiple Instruction Multiple Data (MIMD) architectures.

Null Firmware Null firmware is firmware that has no static rules. Null firmware is used to maximize the dynamic rule capacity on the FPGA.

Offset Offset is a Snort keyword that specifies a pattern-matching start location within a packet. For example, an offset of 5 directs Snort inspect packets beginning after the first 5 bytes of the payload. The P-Series does not support this Snort keyword. Rather, the P-Series has an offset feature that enables offsets for all rules. This feature is optionally activated during the PNIC-Compiler configuration phase.

meta.rules meta.rules is a Snort rules file supplied with the P-Series appliance by Force10. The rules in this file report on flow information and handle possible TCP segmentation evasion attempts. They also provide compatibility with Snort, and including them allows you to run Snort on the DPI interface.

SFP Small Form-factor Pluggable (SFP) is an optical transceiver that interfaces a network device and a fiber or unshielded twisted pair (UTP) network cable. SFPs support the SONET and Gigabit Ethernet standards and can transmit data at a rate of 4.25 Gb/s.

Snort Snort is an open source network intrusion detection and prevention system that uses rules created with a special syntax to examine and control specified traffic.

SPAN Port Switched Port Analyzer (SPAN) Port is a switch port that receives a copy of specific traffic that passes through a switch. The SPAN port is also called a mirroring port.

State State is information about a flow including the source address, destination address, source port, and destination port. See Flow.

Static Rules Static rules are rules that are specified in a file using Snort syntax, and then compiled to become part of the firmware. Static rules can be disabled/enabled individually, but they cannot be changed once they have been loaded into the FPGA. To change static rules, you make changes to the rules in the original rules file, recompile them, and reload the new firmware in the FPGA.

SYN A synchronous packet (SYN) is a packet sent from the client to the server that requests a TCP connection. It is the first part of the TCP handshake that establishes a TCP connection between the client and server.

The second part of the handshake is where the server sends a SYN-ACK packet back to the client to acknowledge the receipt of the SYN request. Finally, the client sends an ACK packet to the server to complete the connection. A SYN flood is a type of denial of service attack where a series of handshakes is initiated but not completed because the final ACK packet is never sent to the server. This occupies the server’s resources, which results in a denial of service for other clients. See ACK.

Tap A tap is a device that can passively monitor network traffic, and is analogous to a telephone wire tap.

XFP XFP is a tranceiver that interfaces a network device and a fiber or unsheilded twisted pair (UTP) network cable. It can transmit data at a rate of 10 Gb/s.

Appendix F Technical Support

Manual Pages

Information on operating the appliance can be accessed through manual pages (man pages) with the command **man** *command.* The command **man pnic** displays the man pages on the command line interface; and **man pnic** displays them on the *Ncurses* interface. Man pages for the compiler can be accessed with **man pnic-compiler**.

- For information on Snort or creating Snort rules, visit www.snort.org.
- For information on Unix commands and the *vi* editor, see Appendix D , on page 125.

The iSupport Website

iSupport provides a range of documents and tools to assist you with effectively using Force10 equipment and mitigating the impact of network outages. Through iSupport you can obtain technical information regarding Force10 products, access to software upgrades and patches, and open and manage your Technical Assistance Center (TAC) cases. Force10 iSupport provides integrated, secure access to these services.

Accessing iSupport Services

The URL for iSupport is www.force10networks.com/support/. To access iSupport services you must have a userid and password. If you do not have one, you can request one at the website:

1. On the Force10 Networks iSupport page, click the **Account Request** link.
2. Fill out the User Account Request form, and click **Send**. You will receive your userid and password by E-Mail.
3. To access iSupport services, click the **Log in** link, and enter your userid and password.

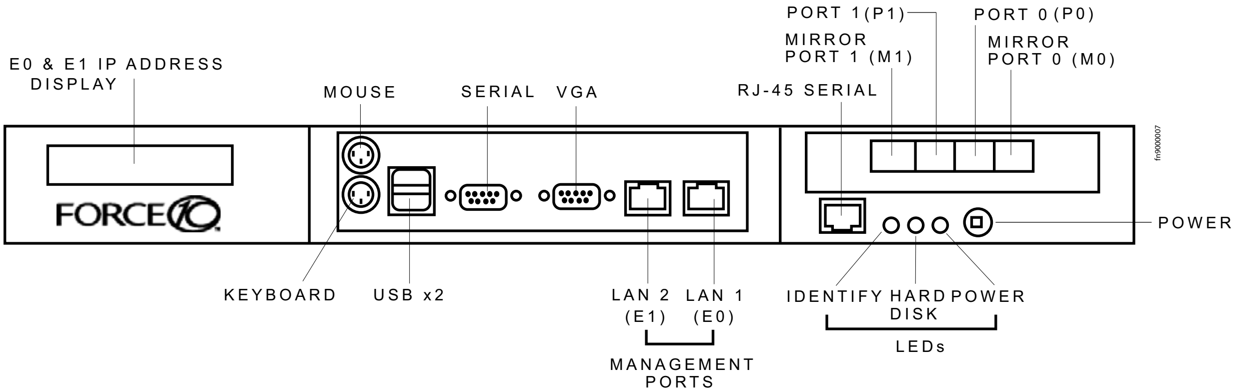
Contacting the Technical Assistance Center

How to Contact Force10 TAC	Log in to iSupport at www.force10networks.com/support/, and select the **Service Request** tab.
Information to Submit When Opening a Support Case	• Your name, company name, phone number, and E-mail address • Preferred method of contact • Model number • Serial Number (see Locating P-Series Serial Numbers on page 130) • Software version number • Symptom description • Screen shots illustrating the symptom, including any error messages.
Managing Your Case	Log in to iSupport, and select the **Service Request** tab to view all open cases and RMAs.
Downloading Software Updates	Log in to iSupport, and select the **Software Center** tab.
Technical Documentation	Log in to iSupport, and select the **Documents** tab. This page can be accessed without logging in via the **Documentation** link on the iSupport page.
Contact Information	E-mail: support@force10networks.com Web: www.force10networks.com/support/ Telephone: US and Canada: 866.965.5800 International: 408.965.5800

Locating P-Series Serial Numbers

The P10 serial number is located on a sticker on the back of the unit in the top-right corner (see Figure 2), as well as on the left mounting bracket (see Figure 87). The serial number is below the bar code and has 8 characters.

Figure 87 Location of P10 Serial Number



Requesting a Hardware Replacement

To request replacement hardware, follow these steps:

Step	Task
1	Determine the part number and serial number of the component.
2	Request a Return Materials Authorization (RMA) number from TAC by opening a support case. Open a support case by: • Using the Create Service Request form on the iSupport page (see Contacting the Technical Assistance Center on page 130). • Contacting Force10 directly by E-mail or by phone (see Contacting the Technical Assistance Center on page 130). Provide the following information when using E-mail or phone: • Part number, description, and serial number of the component. • Your name, organization name, telephone number, fax number, and E-mail address. • Shipping address for the replacement component, including a contact name, phone number, and E-mail address. • A description of the failure, including error messages. • The support representative will validate your request and issue an RMA number for the return of the component.
3	Pack the component for shipment. Label the package with the component RMA number.